

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ І.А. Дичка

«__» _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

на тему: «Веб-платформа для розміщення реклами. Клієнтська частина»

Виконав:

студент IV курсу, групи КП-52

Сорочинський Владислав Володимирович

Керівник:

Ст. викладач кафедри ПЗКС, к.т.н.,

Рибачок Н.А.

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н.,

Онай М.В.

Рецензент:

Доцент кафедри ЕІ ФЕЛ, к.т.н.,

Вунтесмері Ю.В.

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря
Сікорського”**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050103 “Програмна інженерія”

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Сорочинському Владиславу Володимировичу

1. Тема проекту «Веб-платформа для розміщення реклами. Клієнтська частина», керівник проекту Рибачок Наталія Антонівна, к.т.н., старший викладач, затверджені наказом по університету від “22” травня 2019 року №1331-С
2. Термін подання студентом проекту: “20” червня 2019 р.
3. Вихідні дані для дипломного проектування: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - розробити загальну структуру клієнтської частини сервісу;
 - визначити організацію інформації на кожній сторінці сервісу;
 - виконати програмну реалізацію веб-сервісу відповідно до вимог технічного завдання;
 - виконати тестування веб-сервісу.
5. Перелік обов’язкового ілюстративного матеріалу:
 - схема авторизації користувача (креслення);
 - граф переходів веб-додатку (креслення);
 - структура клієнтського веб-додатку React (плакат);
 - архітектура розробленого веб-додатку (плакат).
6. Консультанти:

| Питання | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Онай М.В., старший викладач | | |

7. Дата видачі завдання: “31” жовтня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломного проекту | Строк виконання етапів | Примітка |
|-------|---|------------------------|----------|
| 1. | Вивчення літератури за тематикою проекту | 29.11.2018 | |
| 2. | Розроблення та узгодження технічного завдання | 29.11.2018 | |
| 3. | Розроблення структури інформаційної системи | 16.12.2018 | |
| 4. | Підготовка матеріалів першого розділу дипломного проекту | 18.01.2019 | |
| 5. | Підготовка матеріалів другого розділу дипломного проекту | 30.01.2019 | |
| 6. | Програмна реалізація інформаційної системи | 07.04.2019 | |
| 7. | Тестування інформаційної системи | 27.04.2019 | |
| 8. | Підготовка матеріалів третього розділу дипломного проекту | 05.05.2019 | |
| 9. | Підготовка матеріалів четвертого розділу дипломного проекту | 21.05.2019 | |
| 10. | Підготовка графічної частини дипломного проекту | 01.06.2019 | |
| 11. | Оформлення документації дипломного проекту | 08.06.2019 | |

Студент _____ **Сорочинський В.В.**

Керівник проекту _____ **Рибачок Н.А.**

ВЕБ-ПЛАТФОРМА ДЛЯ РОЗМІЩЕННЯ РЕКЛАМИ. КЛІЄНТСЬКА ЧАСТИНА

АНОТАЦІЯ

Даний дипломний проект присвячений створенню програмного забезпечення для клієнтської частини рекламної веб-платформи.

Інтерактивна інформаційна система являє собою односторінковий web-додаток. Для зареєстрованого користувача доступні функції перегляду статистики, створення, редагування та керування рекламними кампаніями, банерами. Після отримання статусу Видавця стають доступні сторінки керування рекламними блоками. Сайт передбачає лише авторизований доступ, відтак гостьовий перегляд можливий лише для сторінок реєстрації та входу. Після проходження авторизації користувач може поповнювати свій аккаунт за допомогою біткоїн-гаманця та направляти кошти на фінансування конкретних кампаній.

У даному дипломному проекті розроблено: архітектуру інформаційної системи, алгоритм авторизації, функціональність керування маркетинговими кампаніями та також графічні елементи та дизайн web-сторінок.

WEB-PLATFORM FOR ADVERTISING. FRONT END PART

ABSTRACT

This diploma project is dedicated to creating a software for front end part of the marketing web-platform.

The interactive information system is a single-web-application. An authorized user has the access to viewing statistics, creating, updating and managing marketing campaigns and banners. After receiving a status of Publisher he can also visit pages for advertisement blocks management. Web-site provides authenticated access only, so unapproved users can only surf through sign up and sign in pages. After going through authorization user can replenish his account using bitcoin transaction and direct funds to finance particular campaigns.

During this project, it was developed the following: information system architecture, algorithm of authorization, functionality for managing marketing campaigns and also graphic elements and web-pages design.

ДП.045440-01-90 Веб-платформа для розміщення реклами. Клієнтська частина. Відомість проекту

| Позначення | Найменування | Кі-л-ть | Примітка |
|-----------------|-------------------------|---------|----------|
| | Документація проекту | | |
| | | | |
| ДП.045440-02-91 | Веб-платформа | 4 | |
| | для розміщення реклами. | | |
| | Клієнтська частина. | | |
| | Технічне завдання | | |
| | | | |
| ДП.045440-03-81 | Веб-платформа | 51 | |
| | для розміщення реклами. | | |
| | Клієнтська частина. | | |
| | Пояснювальна записка | | |
| | | | |
| ДП.045440-04-51 | Веб-платформа | 4 | |
| | для розміщення реклами. | | |
| | Клієнтська частина. | | |
| | Програма та методика | | |
| | тестування | | |
| | | | |
| ДП.045440-05-34 | Веб-платформа | 6 | |
| | для розміщення реклами. | | |
| | Клієнтська частина. | | |
| | Керівництво користувача | | |
| | | | |
| ДП.045440-06-99 | Веб-платформа | 1 | |
| | для розміщення реклами. | | |
| | Клієнтська частина. | | |
| | Схема авторизації | | |
| | додатку. UML-діаграма | | |
| | послідовності | | |
| | | | |
| | | | |
| | | | |

ДП.045440-01-90 Веб-платформа для розміщення реклами. Клієнтська частина.
Відомість проекту

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“__” _____ 2018 р.

ВЕБ-ПЛАТФОРМА ДЛЯ РОЗМІЩЕННЯ РЕКЛАМИ.
КЛІЄНТСЬКА ЧАСТИНА

Технічне завдання

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проекту:

_____ Н.А. Рибачок

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ В.В. Сорочинський

ЗМІСТ

| | |
|---|---|
| 1. Найменування та галузь застосування..... | 3 |
| 2. Підстава для розроблення | 3 |
| 3. Призначення розробки | 3 |
| 4. Вимоги до програмного продукту | 3 |
| 5. Вимоги до проектної документації | 4 |
| 6. Етапи проектування | 4 |
| 7. Порядок тестування розробки | 4 |

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Веб-платформа для розміщення реклами.

Клієнтська частина.

Галузь застосування: маркетинг, продажі.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розроблювана інформаційна система призначена для використання в маркетингу для просування товарів та послуг.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Даний програмний продукт повинен виконувати такі основні функції:

1. Рекламодавець має можливість створювати рекламні кампанії, групи та конкретні банери.
2. Рекламодавець має можливість переглядати статистику зазначених сутностей.
3. Рекламодавець може поповнювати баланс свого аккаунту за допомогою біткоїн-гаманця.
4. Рекламодавець може керувати, призупиняти, встановлювати фінансовий ліміт маркетингових кампаній.
5. Видавець може додавати власні сайти для відображення реклами інших користувачів, розміщати на них маркетингові блоки, керувати ними та переглядати детальну статистику.

Додаткові вимоги:

1. Кросбраузерність та кросплатформеність.
2. Зручний у користуванні та зрозумілий інтерфейс.

5. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проекту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Схема авторизації користувача»;
 - «Граф переходів веб-додатку».
- 5) плакати:
 - «Структура клієнтського веб-додатку React»;
 - «Архітектура розробленого веб-додатку».

6. ЕТАПИ ПРОЕКТУВАННЯ

| | |
|---|------------|
| Вивчення літератури за тематикою проекту | 29.11.2018 |
| Розроблення та узгодження технічного завдання | 29.11.2018 |
| Розроблення структури інформаційної системи | 16.12.2018 |
| Програмна реалізація інформаційної системи | 07.04.2019 |
| Тестування інформаційної системи | 27.04.2019 |
| Підготовка матеріалів текстової частини проекту | 21.05.2019 |
| Підготовка графічної частини дипломного проекту | 01.06.2019 |
| Оформлення документації дипломного проекту | 08.06.2019 |

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ І.А. Дичка

« ____ » _____ 2019 р.

ВЕБ-ПЛАТФОРМА ДЛЯ РОЗМІЩЕННЯ РЕКЛАМИ. КЛІЄНТСЬКА
ЧАСТИНА

Пояснювальна записка

ДП.045440-03-81

«ПОГОДЖЕНО»

Керівник проекту:

_____ Н.А. Рибачок

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ В.В. Сорочинський

ЗМІСТ

| | |
|--|----|
| СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ | 3 |
| ВСТУП..... | 5 |
| РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ | 6 |
| 1.1. Обґрунтування вибору критеріїв оцінювання програмних засобів | 6 |
| 1.2. Аналіз існуючих рішень для інтернет-маркетингу..... | 7 |
| 1.3. Висновок | 10 |
| РОЗДІЛ 2. ОБґРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ | 12 |
| 2.1. Обґрунтування вибору типу програмного забезпечення | 12 |
| 2.2. Вибір мови програмування для розроблення клієнтської частини веб-додатку | 16 |
| 2.3. Вибір фреймворку для розроблення клієнтської частини веб-додатку | 23 |
| РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ВЕБ-ДОДАТКУ..... | 30 |
| 3.1. Загальний опис веб-додатку | 30 |
| 3.2. Функціональні вимоги до веб-додатку..... | 31 |
| 3.3. Архітектурна веб-додатку..... | 32 |
| 3.4. Висновок | 35 |
| РОЗДІЛ 4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБ-ДОДАТКУ | 36 |
| 4.1. Робочий потік веб-додатку | 36 |
| 4.2. Тестування веб-додатку | 45 |
| 4.3. Порівняння розробки з існуючими аналогами..... | 46 |
| 4.4. Рекомендації щодо подальшого вдосконалення | 47 |
| ВИСНОВКИ | 49 |
| СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ..... | 50 |
| ДОДАТКИ | 51 |

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БД – база даних, зберігає дані, надає їх серверній частині за запитом.
Бандлер – програмний скрипт, що збирає усі модулі в єдиний файл, щоб уникнути необхідності завантаження декількох модулів у інтернет браузер користувача.

Видавець – сайт, що розміщує рекламу, надану розроблюваною рекламною веб-платформою.

Генерація лідів – процес залучення потенційних покупців.

Десктоп (desktop) – платформа для роботи програмного додатку, персональний комп'ютер або ноутбук.

Емуляція – імітація потрібного середовища в межах існуючого.

Конверсія – відношення кількості відвідувачів сайту або тих, хто переглянув інформацію про послугу до тих, хто здійснив будь-які дії (для веб-додатку це кліки, переходи на нові сторінки в межах застосунку, а для товару - придбання).

Лід – особа, яка є потенційним покупцем.

Маркетинг – система заходів для просування певної послуги, продукту, сервісу, події тощо.

Плагін – частина коду третьої сторони, що вставляється на сайт або інтернет-браузер для виконання певної задачі: програвання відео, зображення контенту тощо.

Показ – відображення рекламного банера в інформаційному просторі платформи, що потрапляє в поле зору користувача.

ПЗ – програмне забезпечення.

Рекламодавець – користувач, що купує право розміщувати рекламу за допомогою розроблюваної платформи.

Таргетинг – набір опцій для демонстрації реклами певній аудиторії, наприклад, певним соціальним, мовним групам тощо.

Товар – рекламована послуга або предмет продажу.

Back end – серверна частина додатку, надає дані та здійснює зміни в базі даних за запитом.

Front end – клієнтська частина додатку, що відповідає за візуальне представлення даних та надає інтерфейс для маніпуляцій з ними.

CPC (cost per click) – тарифікаційний план рекламних сервісів, за якого користувач здійснює оплату за кількість переходів за посиланням, тобто клік по демонстрованому банеру.

CPI (cost per impression) – тарифікаційний план рекламних сервісів, що означає оплату за кількість показів банера.

CTR (click-through rate) – відношення кількості переходів за посиланням до частоти показів рекламного банера.

SEO (search engine optimization) – оптимізація пошукових систем.

UI (user interface) – інтерфейс користувача, набір візуальних елементів, що надається користувачу для використання функцій додатку.

UX (user experience) – зручність використання додатку для кінцевого користувача.

ВСТУП

Інтернет-маркетинг – практика використання стратегій традиційного маркетингу у інтернет-просторі з метою просування продукту або послуг для керування трафіком, генерації продажів та лідів. Включає в себе контентну рекламу, розсилку електронною поштою, пошукову рекламу, платні маркетингові майданчики тощо. Він є дешевшим через відсутність обов'язкової фізичної присутності рекламного об'єкту (наприклад, банера).

Зараз переважну частину інформаційного простору для інтернет-маркетингу контролюють великі корпорації, на кшталт Apple, Facebook та Google. Перша фокусується переважно на рекламі у мобільних додатках, останні дві, окрім цього, також охоплюють медіа простір на веб сайтах.

Ці корпорації не тільки мають жорсткі правила стосовно рекламованих продуктів, але і задають ці стандарти меншим компаніям. Відповідно рекламодавці, що надають специфічні товари і послуги (трейдинг криптовалют, азартні ігри), не можуть охопити свою аудиторію і збільшити продажі та конверсію, оскільки немає платформи, політика якої дозволяє їм розвертати свої маркетингові кампанії.

Види діяльності, що підпадають під обмеження контентної політики великих рекламних платформ:

- майнінг біткоінів;
- інвестування та обмін біткоінів;
- форуми, блоги про криптовалюту;
- онлайн казино.

Основним завданням дипломного проекту є розробка програмного забезпечення для клієнтської частини веб-платформи для розміщення реклами. Підставою для такої розробки стала проблема монополізації інтернет-маркетингу.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

Перш за все сформуємо основні потреби клієнтської частини веб-платформи для розміщення реклами. До системи було висунуто наступні вимоги:

- можливість надавати рекламні місця на своєму сайті в оренду;
- можливість розміщувати свої медіа-матеріали (рекламу);
- гнучка контентна політика;
- кросплатформеність;
- можливість оплати послуг за допомогою біткоїну ;
- конфіденційність та захист особистих даних;
- можливість перегляду статистики.

1.1. Обґрунтування вибору критеріїв оцінювання програмних засобів

В результаті подальшого аналізу вимог до додатку було визначено наступні функціональні вимоги:

1. Можливість оплати послуг платформи за допомогою біткоїн-гаманця.
2. Можливість налаштовувати таргетинг.
3. Гнучкі налаштування формату реклами – можливість обрати текст, анімацію, відео різних розмірів.
4. Гнучка тарифікація – можливість вибрати різні режими оплати, зокрема за кількість показів реклами або кількість переходів за посиланням.
5. Можливість перегляду детальної статистики за рекламними кампаніями та окремими банерами.
6. Фіксована ціна на послуги платформи, що не буде перераховуватися.

7. Можливість залишити заявку на те, щоб стати Видавцем (тобто право надавати рекламні місця на своєму сайті для демонстрації маркетингових матеріалів платформи).

Також було визначено нефункціональні вимоги:

1. Стабільна робота на будь-якій з наступних платформ: Windows, Linux, Mac OS, Android, IOS. Представлені в даному списку операційні системи є найпоширенішими, тому комфортна робота користувачів додатку в цих середовищах надзвичайно важлива.
2. Інтуїтивність та простота інтерфейсу платформи – чітке візуальне розділення частин системи на кампанії, що складаються з рекламних банерів.
3. Блокування елементів інтерфейсу під час проведення запитів до серверної частини для уникнення помилок зі сторони користувача.

1.2. Аналіз існуючих рішень для інтернет-маркетингу

Проаналізувавши існуючі рішення для інтернет-маркетингу на основі визначених вище критеріїв оцінювання програмних засобів, можна помітити, що кожен з них має свої недоліки. Переважно це пов'язано з обмеженнями сервісів та нестабільною ціною на власні послуги. Нижче представлені найпотужніші рекламні платформи, наведено їх переваги та недоліки.

1.2.1. Google AdWords

Google Ads – онлайн маркетингова платформа, розроблена компанією Google, що надає компаніям та підприємцям послуги з демонстрації коротких текстів, невеликого каталогу товарів у вигляді сітки, сервісних пропозицій, відео контенту та генерації рекламних матеріалів у мобільних додатках для користувачів у межах мережі Google.

Рекламна платформа базується на даних кешу браузера користувача та ключових словах, визначених рекламодавцем. Система аналізує цю інформацію і розміщає об'яву на сторінках для користувачів, яким вона може бути цікава [1, 2].

Налаштування таргетингу сервісу включають в себе можливість концентрації кампанії на конкретних доменах, демографічних та національних групах. Також рекламодавцю дозволяється додавати виключення за цими ж самими принципами.

Тарифікаційна модель платформи – оплата за кліки. Проте ціна за послуги може зростати, якщо в цьому ж сегменті діяльності бізнесу інші користувачі купують рекламні місця.

Контентна політика має значні обмеження, що варіюються в залежності від країни. Переважно заборонені азартні ігри (казино, автомати, лотереї), алкоголь, сервіси, що стосуються будь-яких криптовалют та операцій з ними, сайти знайомств також класифікуються як дорослий контент (з червня 2014 року по всьому світу).

Статистика надається за кожною рекламною кампанією окремо та за акаунтом користувача в цілому.

До платформи можна приєднатися в якості Видавця. Для цього існує окремий функціональність Google AdSense.

Загалом переваги використання сервісу включають:

- швидкі результати зі збільшення трафіку;
- SEO оптимізацію проводить Google, що економить ресурси компанії;
- надання детальної статистики;
- велике інформаційне поле для відображення рекламних матеріалів (найбільша пошукова система світу);
- швидке створення та початкове налаштування кампанії.

До недоліків платформи відносяться:

- зростання цін для підприємницьких сегментів з високою конкуренцією;
- складний інтерфейс, що потребує часу для опанування;
- сувора контентна політика;
- обмежена кількість символів для рекламного повідомлення (3 заголовки по 30 символів кожен);
- відсутність біткоіну в якості способу оплати.

1.2.2. Facebook Ads Manager

Facebook Ads Manager – рекламний сервіс, що надає соціальна мережа Facebook.

Місцем для проведення кампанії в даному випадку є стрічка новин користувача цієї соціальної мережі та інших продуктів, розроблених цією корпорацією (наприклад, Instagram).

Сервіс надає дуже широкі налаштування таргетингу – найбільші серед доступних платформ. Вони включають в себе географічне положення користувача, стать, вік, місце роботи, чи знаходиться він у стосунках тощо. Також враховуються його інтереси на основі аналізу постів, сторінок, які він вподобав, товарів, які придбав, та інших активностей. Це надає інструменту величезний потенціал, але вимагає від користувача ґрунтовних знань з маркетингу.

Платформа також має гнучку тарифікацію. Оплата може здійснюватися за показ або за клік. Її не можна здійснити за допомогою криптовалюти.

Контента політика має аналогічні до Google AdWords обмеження. Окрім того має визначені стандарти формату реклами, зокрема не більш ніж 20 відсотків тексту на зображенні [3].

В результаті аналізу маємо наступні переваги:

- низька вартість;

- простий у використанні інтерфейс для керування налаштуваннями кампанії;
- великий обсяг документації, що допомагає у використанні сервісу.

Недоліками сервісу є:

- необхідні знання з маркетингу, без яких використання платформи може бути неефективним;
- складний процес схвалення рекламних кампаній;
- відсутність підтримки оплати своїх послуг біткоіном.

1.3. Висновок

Після огляду існуючих рішень для інтернет-маркетингу можна зробити висновок, що жодне з них не задовольняє критеріям, сформульованим у розділі 1 (див. табл. 1.1) коротко зібрана інформація стосовно відповідності аналогів попередньо визначеним вимогам.

Таблиця 1.1

Відповідність існуючих рішень критеріям оцінювання

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------------|----|----------|-----|-----|----------|-----|-----|
| Google AdWords | Ні | Частково | Так | Ні | Так | Ні | Так |
| Facebook Ads Manager | Ні | Так | Ні | Так | Частково | Так | Ні |

З порівняльної таблиці бачимо, що жоден сервіс не надає достатньо ліберальної політики для використання своїх платформ деякою кількістю компаній та підприємців.

Хоча Facebook Ads Manager надає надзвичайно варіативні налаштування таргетингу, вони потребують ґрунтовної освітньої бази

з маркетингового напрямку, а інформаційний простір, що використовує платформа, значно обмежує формат маркетингової діяльності в цьому сервісі.

Загалом Google AdWords надає більшу функціональність, проте також недоліки, а саме нефіксовану ціну на свої послуги, обмеження у планах тарифікації та варіативності формату рекламних матеріалів. Та незважаючи на це, можливість співпраці та монетизації веб-сайтів роблять його більш привабливим для Видавців, аніж Facebook Ads Manager.

Обидві платформи мають власні обмеження з формату маркетингового контенту, і, відповідно, мають свою аудиторію користувачів, проте не вирішують проблеми обмежень контентної політики у інформаційному онлайн просторі для маркетингу та не враховують тренд на біткоїн (відповідно, не задовольняють зростаючий попит на нього).

Таким чином, було прийнято рішення про розробку веб-платформи для розміщення реклами, що відповідатиме зазначеним вище вимогам та враховуватиме недоліки існуючих програмних рішень.

РОЗДІЛ 2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Обґрунтування вибору типу програмного забезпечення

З аналізу основних вимог до рекламної платформи, зроблено висновок, що розробка повинна проводитися в межах системної архітектури клієнт-сервер: клієнтська частина додатку відображає користувачу візуалізовані дані, що отримуються з бекенду за допомогою асинхронного запиту до прикладного програмного інтерфейсу. Так будь-які дії з запису, оновлення тощо проводяться тим самим чином. Серверна частина в цьому випадку є посередником для доступу до бази даних для клієнтської. Схема роботи такого застосунку представлена на рис. 2.1.

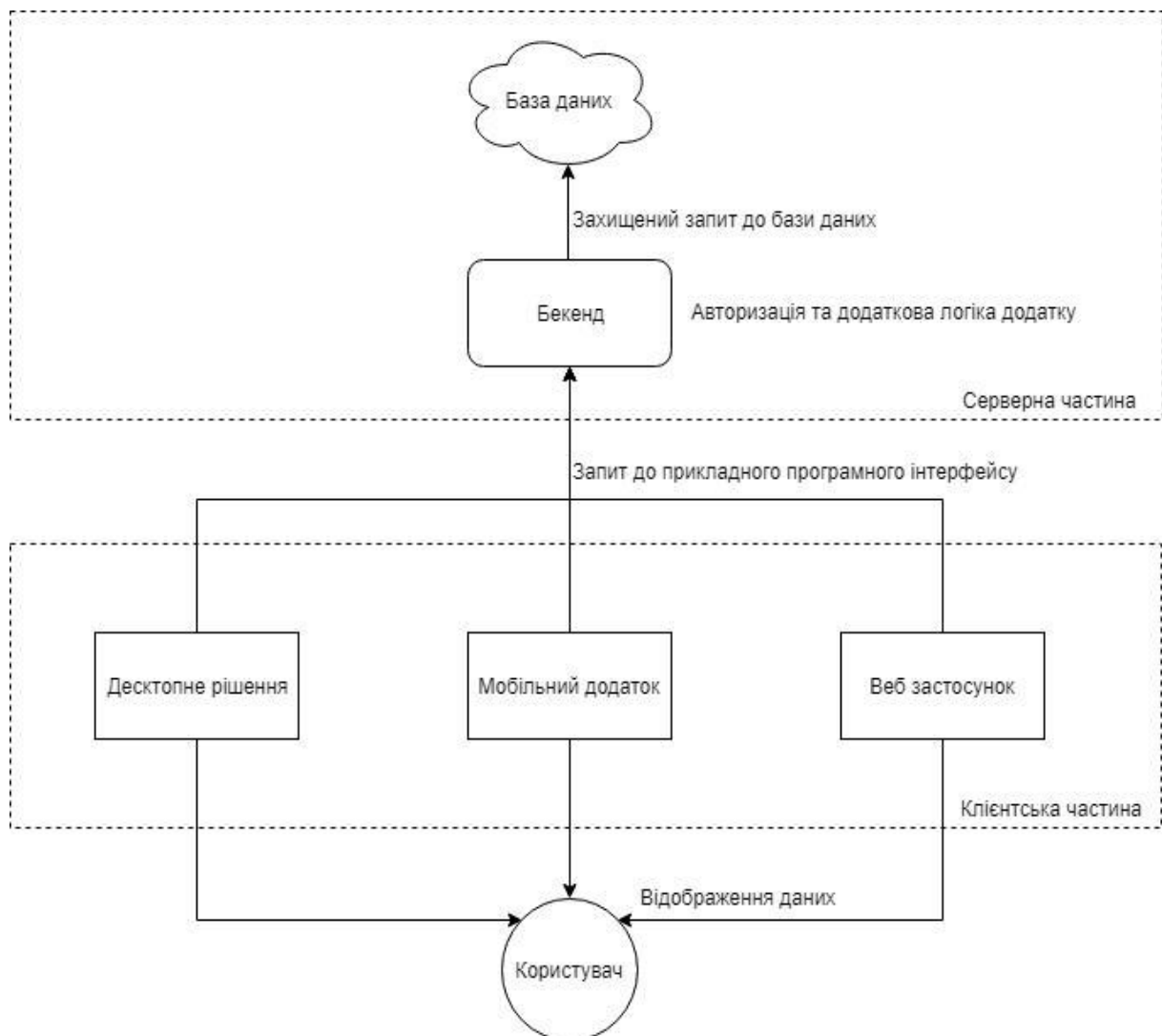


Рис. 2.1. Схема роботи додатку з клієнт-серверною архітектурою

Для вибору засобів реалізації клієнтської частини рекламної платформи, перш за все визначимо тип розроблюваного програмного забезпечення. Для цього сформуємо основні види користувацьких додатків, що задовольняють такій архітектурі застосунку:

1. Комп'ютерний додаток – програмне рішення, що працює на настільному персональному комп'ютері або ноутбуці. Є специфічним для кожної з операційних систем, які можуть використовуватися на зазначених пристроях. Найпоширеніші з них Windows, Linux та MacOS.
2. Веб-додаток – програмне рішення, що використовує інтернет браузер користувача для своєї роботи. Специфікою розроблення подібних застосунків є велика кількість існуючих програм для відображення HTML-сторінок, що мають різні рушії для рендерингу, які розвиваються нерівномірно, проте використовуються практично в однаковій мірі користувачами. Серед головних можна виділити Google Chrome, Opera, Firefox та Safari [4].
3. Мобільний додаток – програмне рішення, що працює на компактних пристроях, а саме смартфонах, планшетах. Розробка є специфічною, оскільки дві основні операційні системи (IOS, розроблений компанією Apple, та Android, який розвиває корпорація Google) є практично рівно розподіленими серед користувачів.

В результаті аналізу головних типів реалізації клієнтських частин користувацьких додатків для розробки рекламної платформи з урахуванням їхніх переваг та недоліків можемо сформулювати порівняльну таблицю, де узагальнимо наявні варіанти (табл. 2.1). Загальні критерії для порівняння:

- поріг входу (необхідність встановлення, налаштування додатку та необхідних інструментів для нього);

- кроссплатформеність (необхідність написання додаткової кодової бази для забезпечення роботи у різних операційних системах);
- потреба в постійних ресурсах (чи резервує платформа на постійній основі ресурси системи, в якій працює);
- потреба в тимчасових ресурсах (скільки потужностей пристрою використовує додаток під час роботи);
- тривалість розробки.

Таблиця 2.1

Порівняльна характеристика клієнтських частин користувацьких додатків

| Ознака | Комп'ютерний додаток | Веб-додаток | Мобільний додаток |
|-------------|---|--|--|
| Поріг входу | Високий. Необхідне встановлення програмного забезпечення на комп'ютер користувача – процес, що може варіюватися в залежності від операційної системи пристрою. | Низький. Для роботи додатку необхідний лише веб браузер у користувача, що є за замовчуванням є частиною кожної сучасної операційної системи. | Середній. Процес встановлення застосунку однаковий для обох платформ IOS та Android (хоча завантаження виконується в їхніх окремих майданчиках), але вимагає від користувача наявності облікового запису (для IOS створюється в процесі налаштування телефону, для пристроїв, що працюють на операційній системі Android – необхідне окреме створення) |

| | | | |
|-------------------------------|--|---|--|
| Кросплатформеність | Відсутня. Необхідне написання окремої кодової бази для кожної операційної системи, де буде використовуватись додаток. | Майже повна підтримка. Потребує незначного розширення кодової бази для коректного відображення стилів у всіх головних браузерах, проте сучасні засоби стилізації (LESS, SASS) частково вирішують цю проблему. | Часткова. Її можна досягти, використавши сучасні фронт енд фреймворки для мобільної розробки: React Native або ж Ionic, проте навіть в такому випадку потребує часткового розширення кодової бази нативною мовою в деяких модулях. |
| Потреба в постійних ресурсах | Присутня. Необхідна пам'ять на жорсткому диску комп'ютера. | Необов'язкова. Проте можна надати права на використання кешу браузера для більшої швидкодії. | Присутня. Займає пам'ять мобільного телефону або картки пам'яті (лише для операційної системи Android). |
| Потреба в тимчасових ресурсах | Висока. Необхідна оперативна пам'ять комп'ютера. | Середня. Потрібні ресурси, виділені, інтернет браузеру та місце для кешу. | Висока. Необхідна значна кількість оперативної пам'яті мобільного пристрою та місце для кешу додатку, щоб зберігати тимчасові дані застосунку. |

| | | | |
|---------------------|---|---|---|
| Тривалість розробки | Висока. Потрібна велика кількість інструментів розробки, компілятори. Використання сторонніх бібліотек – складний процес. | Середня. Потребує встановлення базових інструментів. Існують налагодженні сервіси та бандлери для використання зовнішніх модулів. | Вище середньої. Процес розробки схожий із веб-додатком, проте тестування та розвертання проекту вимагає більших зусиль через емуляцію середовищ IOS та Android. |
|---------------------|---|---|---|

Провівши аналітичне порівняння прийнято рішення про реалізацію клієнтської частини рекламної платформи у вигляді веб-додатку. В умовах обмежених ресурсів та часу стабільна робота на всіх платформах, низька вимогливість до потужності комп'ютера та єдина кодова база надають набагато ширші можливості розробки інтерфейсу користувача, а загальноприйнятий стандарт шаблону проектування «front end – back end» для веб-застосунків вигідно виділяє цей тип програмного забезпечення з-поміж інших.

2.2. Вибір мови програмування для розроблення клієнтської частини веб-додатку

Спершу сформуємо вимоги до мови програмування, яку буде використано для реалізації клієнтської частини рекламної веб-платформи:

- можливість програмування в декількох парадигмах для розроблення різних модулів;
- нестрога типізація;
- можливість реалізації головних принципів об'єктно-орієнтованого програмування;
- високий рівень абстракції;

- наявність фреймворків та бібліотек для комплексного розроблення;
- широкий вибір допоміжних модулів;
- багатопоточність;
- інтерпретованість або компіляція на льоту для більш швидкого процесу розроблення.

2.2.1. Мова JavaScript

JavaScript – високорівнева мультипарадигмова мова, що компілюється на ходу, та виконується у веб-браузері. Блоки (функціональні, умовні, класові тощо) обмежуються фігурними дужками. Мова розвивається в межах визначеного міжнародного стандарту ECMAScript [5, 6].

Разом із мовою розмітки HTML та інструментом стилізації CSS JavaScript формує основу сучасного онлайн простору, а усі найбільші та найпотужніші інтернет-браузери містять в собі ядро для виконання його скриптів.

Мова є прототипно-орієнтованою і не містить вбудованої підтримки класів, проте одна з пізніх версій стандарту ECMAScript, а саме 2015 року, надає так званий «синтаксичний цукор» для об'єктно-орієнтованого програмування, ядром якого є поняття функції-класу.

Типізація мови є динамічною або «качковою» (якщо щось плаває, літає та крикає, то це швидше за все качка). Відтак дані змінюють формат і можуть використовуватися по-різному в залежності від потреби. Підтримується явна та неявні приведення до типів. Найчастіше це використовується під час операцій булевого порівняння, конкатенації та інтерполяції рядків.

Мова надає прикладний програмний інтерфейс для доступу до DOM-дерева сторінки та BOM-інтерфейсу вікна та вкладок браузера.

Серед стандартних вбудованих можливостей також присутні методи для роботи з наступними типами даних:

- текст;
- числа (мова також містить набір констант, пов'язаних з основами програмування, наприклад, максимальне велике ціле число);
- масиви, а також хеш-мапи та хеш-набори;
- об'єкти та їхні прототипи;
- функції (що у JavaScript позиціонуються як змінні, відтак з ними можна проводити ті самі операції, що і з іншими типами даних);
- регулярні вирази (надається великий список вбудованих функцій для операцій над ними: порівняння, пошук, тестування тощо).

Для JavaScript також розроблений глобальний менеджер пакетів NPM (node package manager), що дозволяє розробникам створювати та поширювати власні модулі, а іншим завантажувати їх та використовувати у своїх проектах. Він використовується у всіх сучасних програмах, написаних цією мовою. При ініціалізації проекту створюється файл `package.json`, що містить інформацію про завантажені пакети. Це дозволяє не додавати модулі третіх сторін до систем контролю версій, а лише зазначений файл, що значно пришвидшує роботу з ними. Іншим розробникам необхідно за допомогою спеціальної команди локально встановити модулі з `package.json`.

Для пришвидшення завантаження розроблених додатків JavaScript використовує бандлери – програми, що збирають створені модулі у єдиний файл скрипта, який потім підключається до сторінки користувача. Найвідомішим з таких є `webpack`.

Окрім того до мови розроблено багато фреймворків та великих бібліотек, кожна з яких має свою специфіку та направленість:

- React.js, AngularJS, Angular, Vue.js для front end розробки;
- Electron.js для розробки десктопних додатків;
- React Native та Ionic для мобільної розробки;

- Cordova для доступу до прикладного програмного інтерфейсу мобільних пристроїв (робота з камерою, акселератором тощо);
- Express.js для створення back end.

Динамічна обробка тригерів інтерфейсу, доступу до файлової системи, дій користувача, отримання відповідей з back end у тощо відбувається за моделлю циклу подій. Принцип його дії – черга подій. Оскільки для виконання різноманітних операцій доступно чотири потоки (їхню кількість можна змінювати вручну), JavaScript делегує їм ресурсоємні задачі, на кшталт доступу до файлової системи, DOM-дерева, бази даних, запитів до серверу тощо, а скрипт в цей час продовжує роботу.

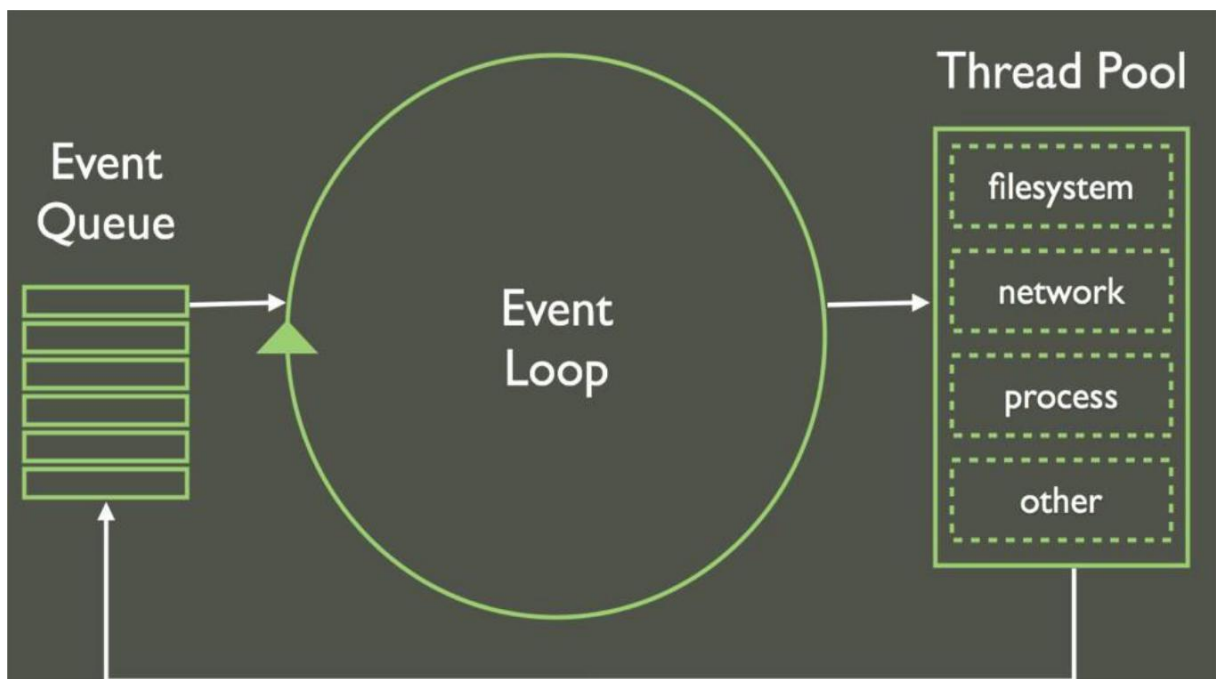


Рис. 2.2. Схема роботи циклу подій JavaScript

Якщо з'являється потреба у виконанні аналогічних операцій, а відповідні потоки зайняті, то ці завдання ставляться в чергу подій, яка перевіряється кожного разу при звільненні необхідного ресурсу. По завершенню операції викликається callback-функція, визначена інженером, з результатами або помилкою в разі невдалого виконання. Схема роботи циклу подій представлена на рис. 2.2.

Окрім синтаксичної обгортки для класів новітній стандарт ECMAScript 2015 приніс інші покращення:

- змінні `let` і `const` локальної області видимості (їхня відмінність полягає у тому, що друга має незмінне значення);
- проміси та синхронні функції – практика очікування виконання `callback`-функції, коли у цьому є необхідність (наприклад, дочекатися відповіді сервера перед відображенням даних);
- анонімні функції, що зберігають контекст в залежності від частини коду, в якій вони визначені, а не тій, де викликаються;
- інтерполяція рядків;
- тип даних `Symbol` – має унікальне значення, яке неможливо зімітувати звичайною строкою;
- генератори – особливі функції, що дозволяють призупиняти свою роботу для отримання проміжних результатів для обробки;
- проксі – об'єктний модуль, який дозволяє перехопити доступ до поля об'єкту при операції читання, запису або редагування.

2.2.2. Мова PHP

PHP (Hypertext processor) – скриптова інтерпретована мова програмування загального призначення, викладена у відкритому доступі. Переважно застосовується для веб-розробки, а саме динамічних сайтів. Мова та її інтерпретатор Zend Engine розробляються групою ентузіастів і є некомерційним проектом [7].

До 2014 року мова розвивалася без визначеного стандарту, роль якого виконувала фактична імплементація.

Синтаксис мови є C-подібним, при цьому частину стандартних можливостей, на кшталт асоціативних масивів, взято з мови Perl. Код вставляється до HTML сторінки за допомогою спеціальних відкриваючого (`<?php`) та закриваючого (`?>`) тегів. Будь-який рендеринг за межами визначеного таким чином блоків є статичним, тобто незмінним. Змінні 20

оголошуються знаком долара (\$), а їхні назви є чутливими до регістру символів. Інструкції розділяються крапкою з комою, як і в більшості мов програмування.

PHP має динамічну типізацію та неявне приведення, що не потребує дій розробника, зокрема під час операцій порівняння. Також існує тип `null`, що означає пустий вказівник – успадкована риса з мови C. Діапазон чисел залежить від платформи, на якій запускається програмний засіб. Підтримуються наступні типи даних:

- ціле число;
- число з плаваючою точкою;
- логічний тип (правда або хиба);
- строка (формат тексту);
- масив;
- об'єкт.

Починаючи з третьої версії PHP частково підтримувала об'єктно-орієнтоване програмування, проте відсутність приватних та захищених полів об'єктів не дозволяла повністю її реалізувати. П'ята версія виправила цей недолік та надала повністю функціональні класи.

Мова має декілька фреймворків, які переважно використовують для розробки серверної частини: `Laravel`, `Symfony`, `Magenta`. Клієнтське розроблення використовує рендеринг з `back end`.

Незважаючи на свої переваги, у PHP відмічається ряд недоліків:

- відсутність двосторонньої підтримки між версіями, що значно ускладнює подальшу підтримку проектів, написаних цією мовою, та обмежує можливості (багато функцій не будуть підтримуватися у майбутньому, як застарілі);
- відсутність багатопоточності – проблема вирішується виключно за допомогою розширень, оскільки стандартні можливості мови не передбачають декількох одночасних процесів при роботі додатку.

2.2.3. Висновок

На основі порівняльної таблиці обираємо JavaScript, як більш стабільну мову програмування, що надає більш широкі можливості з використання зовнішніх пакетів, кращу функціональність, вибір фреймворків для розроблення фронт енду та забезпечує багатопоточність.

Таблиця 2.2

Порівняльна таблиця характеристик мов програмування

| Ознака | JavaScript | PHP |
|--|---|--|
| Можливість мультипарадигмового програмування | Підтримуються об'єктно-орієнтовна, функціональна, імперативна, прототипна парадигми | Підтримується процедурна та об'єктно-орієнтована парадигми |
| Нестрога типізація | Підтримка динамічної та кастоюної типізації | Динамічна типізація |
| Об'єктно орієнтоване програмування | Підтримується | Підтримується |
| Високий рівень абстракції | Присутній | Присутній |
| Широкий вибір допоміжних модулів | Пакетний менеджер npm забезпечує доступ до великої кількості відкритих пакетів | Відсутній єдиний пакетний менеджер, модулі необхідно встановлювати локально |
| Багатопоточність | Досягається за допомогою циклу подій | Можлива за умови використання додаткових бібліотек, проте призводить до зниження швидкодії |

2.3. Вибір фреймворку для розроблення клієнтської частини веб-додатку

Використання чистого JavaScript може призвести до проблем при проектуванні складної клієнтської частини додатку. Вони впливають з необхідності повторної реалізації багатьох базових речей. Це призводить до надмірного витрачання часу та ресурсів. Так, прийнято рішення про використання веб-фреймворку для розробки клієнтської частини.

Веб-фреймворк – це масштабний модуль, що надає широкі стандартні можливості, прикладний програмний інтерфейс для доступу до веб-ресурсів, сервісів, інструментів проектування та розгортання проекту, роутингу, дозволяє інкапсулювати окремі компоненти тощо. Забезпечує високий рівень повторного використання коду [8].

Для реалізації клієнтської частини рекламної веб-платформи сформуємо наступні вимоги до використовуваного фреймворку:

- робота в межах шаблону MVC (модель-відображення-контролер);
- динамічне оновлення даних;
- створення структурної моделі папок при ініціалізації в межах визначеної фреймворком архітектури для мінімізації помилок в процесі розробки;
- проста схема взаємодії з діями користувача;
- двонаправлена передача даних;
- вбудовані можливості для тестування;
- заохочення декларативного стилю програмування для компонентів інтерфейсу та імперативного для логічних сервісів.

2.3.1. React

React.js – бібліотека, розроблена та підтримувана Facebook спільно з окремою групою інженерів та корпорацій, що надає потужності для веб-розробки, близькі до фреймворку. Традиційно застосовується для

побудови потужних односторінкових веб-додатків та мобільних програмних засобів.

Особливістю бібліотеки є використання тіньового DOM-дерева, що є хешованою копією відповідної структури веб-браузера. React використовує його для обчислення різниці між нагальним станом інтерфейсу та наступним. Таким чином браузер оновлює лише частину справжнього DOM-дерева, хоча програмно відображено повністю нову сторінку, підвищуючи швидкодію веб-додатку [9, 10].

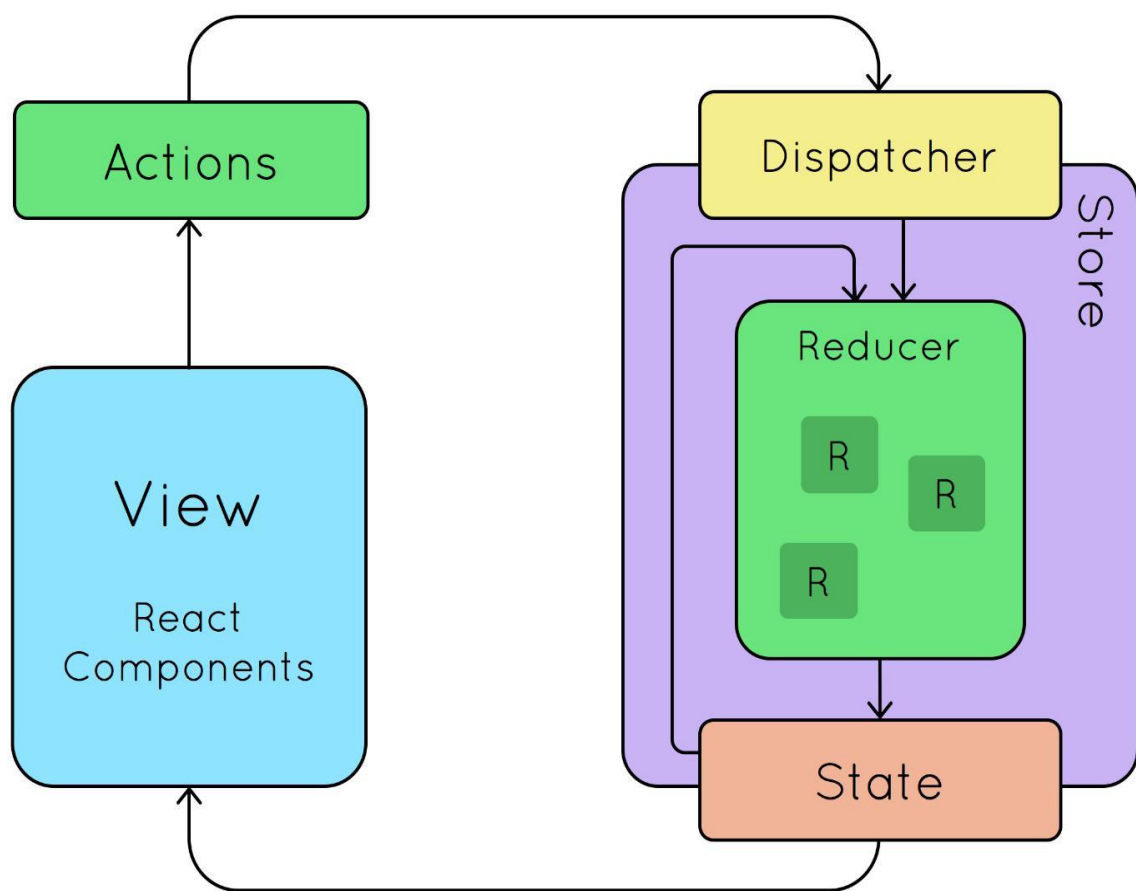


Рис. 2.3. Схема зв'язування даних в React.js

React надає прикладний програмний інтерфейс контексту для керування даними користувача у клієнтській частині веб-застосунку, проте для великих проектів потребує зовнішніх модулів (найчастіше

використовується Redux з проміжними обробниками даних, на кшталт саги). Схема зв'язування даних зображена на рис. 2.3.

Для написання компонентів бібліотека використовує власну модифікацію – JavaScript XML – це мова, що дозволяє використовувати синтаксис схожий до HTML для описання структури користувацького інтерфейсу.

Компоненти React мають локальні (задаються при створенні даної частини інтерфейсу) і глобальні (передані з елемента вищого порядку) дані та власний життєвий цикл, що складається з наступних етапів:

- `render` – відображення компоненту, метод викликається кожного разу при оновленні даних, які використовує елемент;
- `componentDidMount` – метод викликається після поміщення елемента у DOM-дерево сторінки веб-браузера, в ньому традиційно здійснюються запити до сервера на завантаження сутностей для відображення;
- `componentWillUnmount` – викликається перед вилученням елемента з DOM-дерева;
- `shouldComponentUpdate` – викликається перед кожним оновленням і, відповідно, рендерингом компоненту. Метод можна використати для відміни повторного відображення елемента інтерфейсу, повернувши з нього `false`.

Життєві цикли містять більше методів, аніж зазначено, проте решта з них є специфічними і мають вузьку область застосування, зокрема підписка на події оновлення компонентів, присвоєння конкретних значень змінним елементів інтерфейсу.

2.3.2. AngularJS

AngularJS – JavaScript фреймворк для front end розроблення, створений та підтримуваний компанією Google.

Процес написання програмного засобу за допомогою цього фреймворку практично не відрізняється від звичайного HTML, проте має власні унікальні атрибути, які описуються директивами. Вони ж пов'язують розмітку сторінки із інформаційною моделлю, яка являє собою звичайні JavaScript об'єкти.

Фреймворк розроблено з переконанням, що користувацькі та програмні інтерфейси повинні створюватися у декларативному стилі, а сервіси для реалізації бізнес-логіки у імперативному. Стандартний HTML доповнено та адаптовано для реалізації динамічної двонаправленої зв'язки даних для синхронізації різнорівневих частин додатку. Відтак зменшується роль маніпуляцій з DOM-деревом та полегшується тестування коду [11].

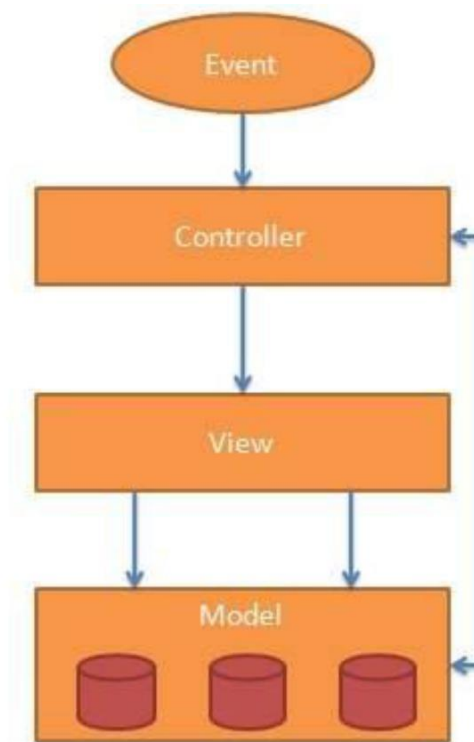


Рис. 2.4. Схема роботи додатку спроектованого за шаблоном MVC

AngularJS при створенні проекту задає розробнику архітектуру додатку MVC (model-view-controller – модель-відображення-контролер). Модель надає дані і змінюється у відповідь на маніпуляції з контролером.

Роль відображення полягає у візуалізації інформації в інтерфейсі програмного засобу. Контролер реагує на дії користувача, сповіщаючи модель про необхідність здійснення змін. Такий підхід дозволяє інкапсулювати дані, операції з ними та їх відображення окремо одне від одного, що зменшує кількість помилок та дозволяє розробнику працювати над різноманітною функціональністю не залежачи від інших членів команди. Схема роботи шаблону представлена на рис. 2.4.

Директиви AngularJS надають наступні можливості:

- оголошення кореневого елемента додатку;
- одностороннє зв'язування даних (зміна значення моделі призводить до зміни тексту в середині елементу відображення);
- двостороннє зв'язування даних (зміна значення в елементі відображення також призводить до зміни значення моделі);
- динамічна зміна класу HTML-елементу;
- створення декількох однакових DOM-елементів для кожної сутності визначеної колекції даних
- відображення або приховування частин інтерфейсу;
- умовний рендеринг елементів (використовується при роутингу, авторизації та розділенні прав користувачів).

2.3.3. *Vue.js*

Vue.js – фреймворк для JavaScript, вихідний код якого знаходиться у відкритому доступі. Головна мета його створення – спрощення побудови користувацьких інтерфейсів та односторонікових веб-застосунків. Vue покладається на декларативний рендеринг та композиційні компоненти.

Синтаксис фреймворку подібний до HTML з деякими модифікаціями. Використовується тіньове DOM-дерево аналогічне до React, але даний модуль сам розраховує мінімально необхідні оновлення справжнього дерева і здійснює їх для зменшення навантаження на інтернет-браузер користувача. Використовується також JavaScript XML, а

одна з останніх версій також повністю інкапсулює рендер всередині компонентів, максимально спрощуючи розробку [12].

Кожен компонент відслідковує змінні, від яких залежить, і ефективно виконує повторне відображення для економії ресурсів браузера. Також вони інкапсулюють код, що повторно використовується.

Перевагою Vue є широкий вибір інструментів для додавання, зміни та видалення компонентів з DOM-дерева. Так, можливе автоматичне застосування анімацій та переходів CSS за допомогою вбудованих стандартних можливостей або бібліотек третіх сторін, маніпуляції з HTML розміткою в процесі зміни стилів. Процес анімування складається з наступних кроків:

1. Перевірка наявності класів анімації у компонента. В разі їх виявлення вони видаляються або додаються у визначений проміжок часу.
2. Якщо анімований компонент має функції-обробники цієї події, їх буде застосовано.
3. У випадку відсутності CSS або скриптових анімацій, операції з DOM-деревом буде проведено негайно в під час наступного оновлення інтерфейсу користувача.

Окрім того надається вбудований стандартний роутинг для усіх ініціалізованих Vue-додатків.

2.3.4. Висновок

В результаті аналізу обираємо фреймворк AngularJS, як єдиний що забезпечує двостороннє зв'язування даних, надає набагато ширшу вбудовану функціональність. Робота в рамках шаблону MVC допомагає дотримуватися вищих стандартів програмування на противагу Vue та React, які не надають жорсткою структури, а відтак вимагають більше досвіду від розробника. Vue також можна було б застосувати для

майбутніх проектів, але для клієнтської частини рекламної веб-платформи його можливостей недостатньо в порівнянні з AngularJS.

РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ВЕБ-ДОДАТКУ

3.1. Загальний опис веб-додатку

Розроблена система є веб-додатком, спроектована за сучасними стандартами користувацьких інтерфейсів і широким набором функціональних можливостей.

Неавторизований користувач не має доступу до додатку. Щоб отримати його він повинен увійти в систему або зареєструватися. Відтак, при початковому завантаженні він бачить лише відповідні форми.

Існує два типи користувачів системи: Рекламодавець та Видавець. При реєстрації кожен, хто входить у систему, отримує право:

- створювати рекламні кампанії і матеріали;
- керувати створеними сутностями;
- поповнювати баланс свого профілю за допомогою біткоіну;
- направляти кошти з фінансового акаунту на фінансування окремих кампаній;
- переглядати статистику.

Для отримання ширших прав, користувач повинен залишити заявку на видавництво. Це передбачено для користувачів, які володіють веб-додатками і хочуть надати розробленій платформі право на розміщення рекламних матеріалів. Для цього заповнюється інформація про тематику діяльності заявленого до видавництва сайту. Після її схвалення користувач стає Видавцем і має право:

- створювати рекламні місця для матеріалів, що надає платформа;
- отримувати фінансовий дохід у біткоінах на баланс свого профілю;
- переглядати статистику рекламних блоків, розміщених на сайті;
- додавати інші сайти для розширення співпраці з платформою.

3.2. Функціональні вимоги до веб-додатку

Неавторизовані користувачі мають право:

- авторизації;
- реєстрації.

Користувачі зі статусом Рекламодавця мають доступ до наступних сторінок та функціональності:

- перегляд загальної статистики за акаунтом;
- створення, видалення, призупинення та поновлення рекламних кампаній;
- поповнення балансу акаунту через біткоїн-транзакцію;
- вивід фінансів з платформи на біткоїн-гаманець;
- переведення коштів профілю на рахунок рекламної кампанії;
- створення рекламного блоку;
- створення рекламного банера, вибір графічного або текстового формату;
- налаштування таргетингу для маркетингових матеріалів;
- редагування профілю;
- перегляд сторінки рекламного блоку;
- перегляд сторінки рекламної кампанії;
- перегляд сторінки рекламного банера;
- перегляд сторінки балансу;
- перегляд сторінки профілю;
- перегляд сторінки заявки на видавництво.

Видавці, крім функцій системи, доступних Рекламодавцю, також мають право:

- створення та редагування блоків для відображення реклами на своєму веб-сайті;
- перегляд статистики рекламних блоків;
- додавання сайтів до свого акаунту;
- деактивацію рекламних блоків;

- перегляду сторінки загальної статистики Видавця (переходи за посиланням, покази реклами, фінансові ресурси отримані за показ реклами);
- перегляду сторінки рекламних блоків;
- перегляду сторінки індивідуальної сторінки блоку для показу реклами з детальною статистикою.

3.3. Архітектура веб-додатку

При розробці клієнтської частини рекламної веб-платформи використано шаблон проектування MVC (Model-View-Controller). Для його реалізації застосовано фреймворк AngularJS.

Структура проекту продиктована необхідністю розділення функціональності системи та відображення інформації щодо сутностей платформи для кожного окремо взятого користувача. Загалом платформа розділена на наступні логічні частини:

1. Статична – включає в себе переважно елементи відображення та стилізації. Для цієї частини виділено окрему папку `static`, яка містить HTML-файли з посиланнями на дані контролера для умовного рендерингу, а також підписки на події оновлення моделі та маніпуляції користувача із програмним інтерфейсом. Окрім того там присутні розроблені файли стилів, зовнішні бібліотеки розмітки та файли ресурсів, а саме зображення, логотипи тощо.
2. Динамічна – включає в себе налаштування та створення додатку AngularJS, підключення зовнішніх модулів, файл та функції, що виконують операції завантаження, оновлення інформації моделі із серверної частини, авторизацію (фактично визначення того, які саме шаблони потрібно відображати), пакет роутингу (вираховує сторінки для рендерингу в інтернет-браузері користувача системи на основі вмісту `url`-рядка) та переходи між сторінками.

Елементи інтерфейсу розділено на чотири основні групи:

1. Шаблони – є статичними частинами клієнтської частини, що відповідають за відображення великих частин системи, функціонально виконують задачі рендерингу, підписки на події, використовують значення моделі, що їм надає контролер.
2. Сторінки – є статичними елементами користувацького інтерфейсу, найвищою ланкою ієрархії роутингу, збирають в собі шаблони для комплексного відображення інформації. Є незмінними в процесі роботи програми.
3. Директиви – найнижча ланка роутингу у AngularJS. Фактично вони є невеликими компонентами, що найтісніше пов'язані з контролером і виконують окремі інкапсульовані завдання системи.
4. Контролери являються файлами JavaScript, що підключені до сторінок та виконують операції надсилення та завантаження з моделлю та сервером. Схема взаємодії їх із моделлю та відображенням представлена на рис. 3.1.

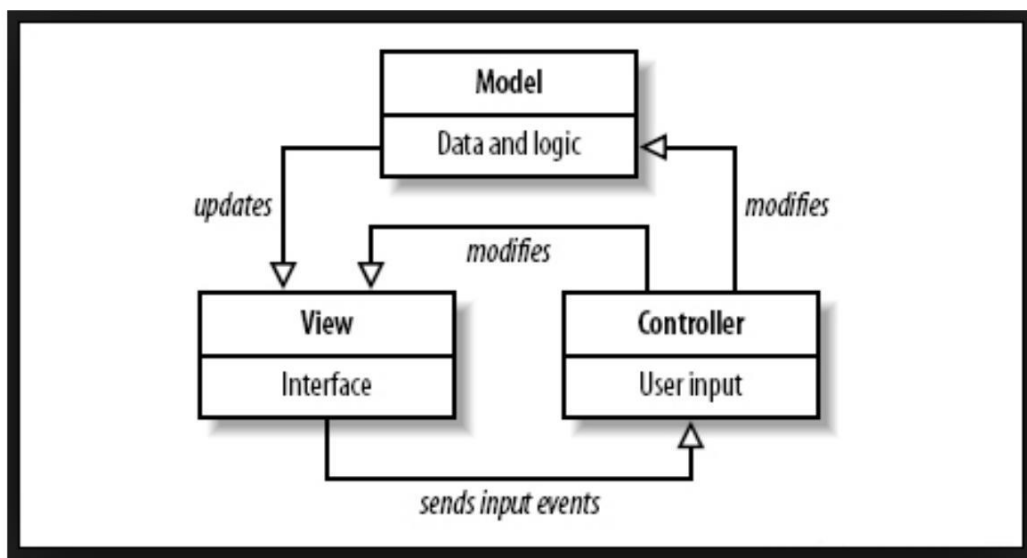


Рис. 3.1. Схема взаємодії частин клієнтської частини веб-додатку за шаблоном MVC

Фільтри – скрипти додатку, що містять функції для роботи з даними. Є посередниками між даними моделі та шаблонами відображення користувацького інтерфейсу застосунку. Містяться в папці `filters` динамічної частини додатку.

Сервіси – скрипти програмного засобу для опрацювання початкових даних в той час, коли справжню інформацію ще не було завантажено з серверної частини платформи, та конфігураційними файлами, що використовуються у зовнішніх пакетах додатку.

Суть взаємодії частин системи полягає у тому, що користувач впливає на контролер, який в свою чергу здійснює маніпуляції з моделлю, що змінює відображення користувацького інтерфейсу. Інкапсуляція елементів платформи, доступна завдяки такому стилю програмування, допомагає їм бути менш залежними одне від одного та зменшити загальну кількість помилок при роботі із системою [13].

Загальну структуру проекту клієнтської частини рекламної веб-платформи наведено на рис. 3.2.

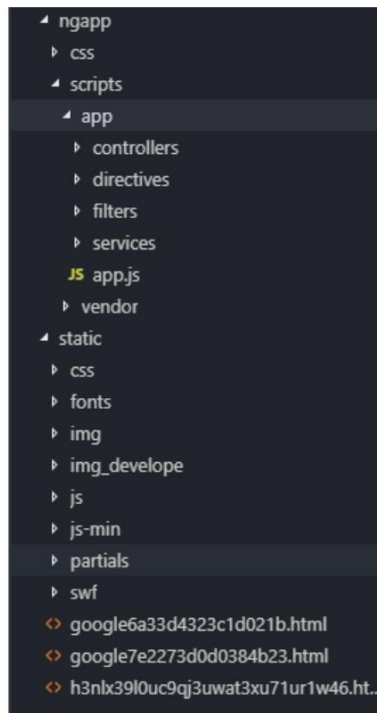


Рис. 3.2. Структура проекту клієнтської частини рекламної веб-платформи

3.4. Висновок

В даному розділі проведено загальний опис розробленої клієнтської частини рекламної веб-платформи, визначено функціональні вимоги, яким повинен задовольняти додаток для неавторизованих користувачів та двох типів зареєстрованих акаунтів: Рекламодавця і Видавця. Дані вимоги повністю реалізовано у розробленому веб-застосунку.

Також розглянуто і описано загальну структуру програмного проекту, а саме Model-View-Controller, що наразі є стандартом для веб-розробки, оскільки забезпечує необхідний рівень інкапсуляції, мінімізацію ймовірності помилок та читабельність коду для подальшої розробки.

Для реалізації вищезгаданого шаблону мовою JavaScript застосовано фреймворк AngularJS.

Для стилізації веб-сторінок підключено зовнішню бібліотеку Bootstrap, що надає зручні класи для роботи з сіткою CSS.

РОЗДІЛ 4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБ-ДОДАТКУ

Систему розроблено в межах шаблону MVC. Робочий потік побудовано за сучасними стандартами проектування робочих інтерфейсів. Загальна схема взаємодії користувача із додатком складається з наступних кроків:

1. Вхід у систему.
2. Завантаження статичного каркасу сторінки.
3. Завантаження окремих елементів користувацького інтерфейсу (директив).
4. Робота контролеру із надсилання запиту на серверну частину для отримання даних про потрібні сутності.
5. Блокування відповідного елемента інтерфейсу для подальшої взаємодії користувача до завершення завантаження інформації про необхідні сутності.
6. Розблокування інтерфейсу користувача для подальших дій, що впливають на програмний стан контролера блоку.

4.1. Робочий потік веб-додатку

На сторінках логіну та реєстрації міститься стандартний для таких форм контент:

- поле вводу для електронної пошти;
- замасковане поле вводу для паролю;
- кнопка підтвердження форми;
- посилання на подібні сторінки додатку (логін, реєстрація, відновлення паролю).

Після входу в додаток користувач перенаправляється на головну сторінку Dashboard. Вони містить короткі відомості про загальну статистику за акаунтом може містити 2 блоки: рекламодавчий та видавничий (відображається в тому разі, якщо профіль було схвалено 36

адміністратором, як Видавець). Користувачу відображаються наступні показники:

1. Витрати (унікальне для рекламодавчого блоку значення) – сума в біткоінах витрачена на рекламні кампанії.
2. Доходи (унікальне для видавничого блоку значення) – сума в біткоінах, зарахована на акаунт за показ реклами інших користувачів на власному сайті.
3. Відношення кількості переходів за посиланням до частоти показів рекламного банера. Є важливим показником для оцінки ефективності кампанії.
4. Кількість показів.
5. Кількість переходів за посиланням.

Наведені вище показники є загальними статистичними поняттями в межах розроблюваної системи і є спільними для усіх сутностей, за якими буде наводитися числова динаміка.

Вкладка Advertiser за замовчуванням демонструє список рекламних кампаній, створених користувачем, наведений у вигляді таблиці, що містить наступні показники за кожною сутністю:

1. Назва кампанії.
2. Кількість рекламних банерів у кампанії.
3. Тип тарифікації: оплата за кількість переходів за посиланням (CPC) або за кількість показів рекламного банера (CPM).
4. Статистика за сьогоднішній день (кількість показів та переходів за посиланням) та за останні сім днів.
5. Статус кампанії: працює, призупинена, проходить перевірку чи відмовлено у публікації.
6. Баланс кампанії у біткоінах (в разі нульового балансу продовження рекламної кампанії неможливе).

Сторінка рекламної кампанії містить загальні відомості про неї (назва, тип тарифікації, баланс) та статистику за банерами, що належать до

неї. Окрім загальних показників (кількість показів, переходів за посиланням та їх відношення), зазначених вище, відображається статус. Числові показники розділені на два періоди: сьогоднішній день та останній тиждень. Нижче розташований графічний блок статистики, він містить наступні елементи:

1. Селектор дат. Доступні визначені наперед значення (один день, два дні, тиждень, місяць, три місяці) та два поля для можливості введення власних у форматі «місяць число рік».
2. Математична крива, що демонструє відношення переходів за посиланням до показів.
3. Графічна крива вартості використання платформи впродовж наданих часових рамок.
4. Зведена таблиця з короткою статистичною інформацією за геолокацією.

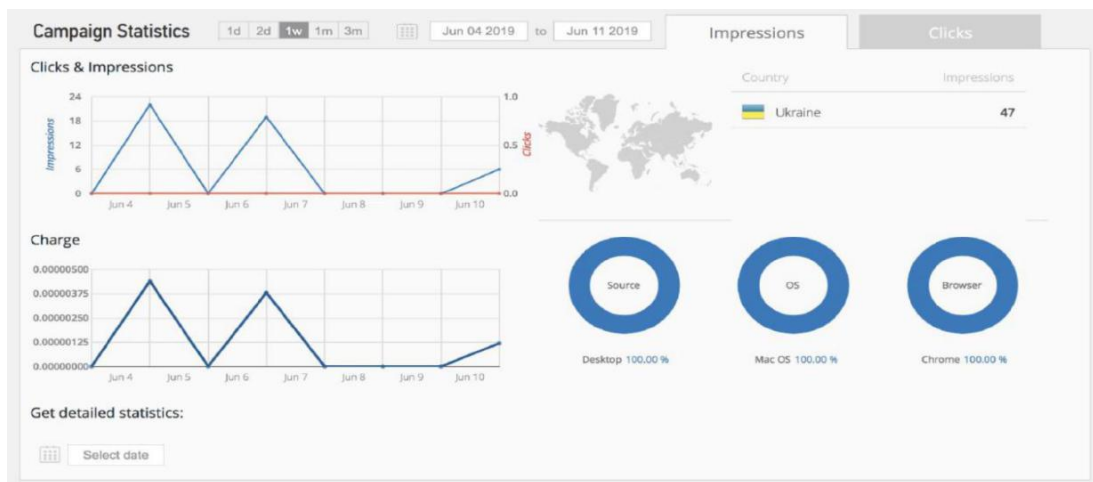


Рис. 4.1. Статистичний графічний блок рекламної кампанії

5. Кругові діаграми зі статистикою рекламної кампанії за наступними налаштуваннями таргетингу: платформа (мобільний телефон, планшет або персональний комп'ютер), операційна система (відповідні для кожного пристрою), інтернет-браузер.

6. Вкладки Impressions та Clicks для переключення режиму відображення показників у зведені таблиці (кількість показів або переходів за посиланням відповідно).

Скріншот статистичного графічного блоку рекламної кампанії зображено на рис. 4.1.

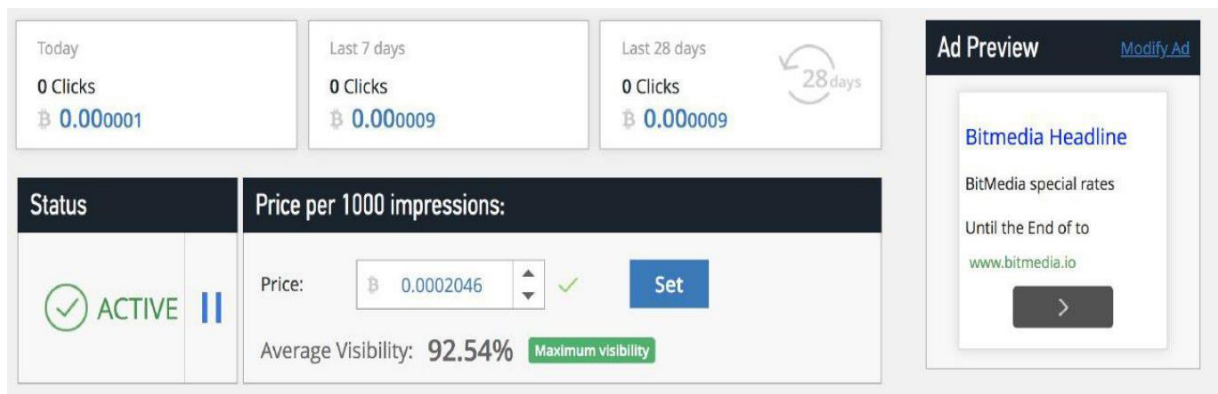


Рис. 4.2. Секція загальної інформації на сторінці рекламного банера

При переході до сторінки рекламного банера відображається вартість переходу за посиланням, щоденний ліміт коштів, що користувач дозволяє витратити зі свого балансу на показ даного маркетингового матеріалу. Вона містить три блоки зі статистикою переходів за посиланням за останній день, тиждень та двадцять вісім днів. Нижче знаходиться блок статусу, де можна призупинити або відновити показ банера, та його демонстраційна версія (як саме він виглядатиме для кінцевого користувача). Верхня секція сторінки представлена рис. 4.2.

Натиснувши на посилання біля моделі перед показу, можна змінити зовнішній вигляд рекламного матеріалу. Для текстових банерів є чотири поля вводу різного формату: заголовок, два текстові та посилання, за яким здійснюється перехід при натиску лівою кнопкою миші. Приклад зображено на рис. 4.3. Графічні банери мають селектор формату, файловий ввід та поле для посилання.

The interface is divided into three main sections:

- Modify Ad:** Contains a 'Text Ad' button.
- Ad details:** A list of ad components with character counts and status icons:
 - Bitmedia Headline: 17/35 ✓
 - BitMedia special rates: 22/35 ✓
 - Until the End of to: 19/35 ✓
 - www.bitmedia.io: 15/35 ✓
 - URL: http:// p2pb2b.io/?utm_source=display ✓
- Ad preview:** A visual representation of the ad showing the headline, rates, and a link to www.bitmedia.io with a right arrow button.

At the bottom, there is a 'Price per 1000 impressions' field set to 0.0002046 and a 'Modify Ad' button.

Рис. 4.3. Сторінка редагування рекламного банера

Вкладка Видавець відображає загальну статистику за веб-сайтами та рекламними майданчиками користувача, що отримав схвалення на видавництво. Зокрема такі показники (скріншот наведено на рис. 4.4):

- кількість показів за сьогоднішній день;
- кількість переходів за посиланням за останню добу;
- дохід за схемою тарифікації оплати за кількість показів рекламних банерів;
- дохід за схемою тарифікації оплати за кількість переходів за посиланням;

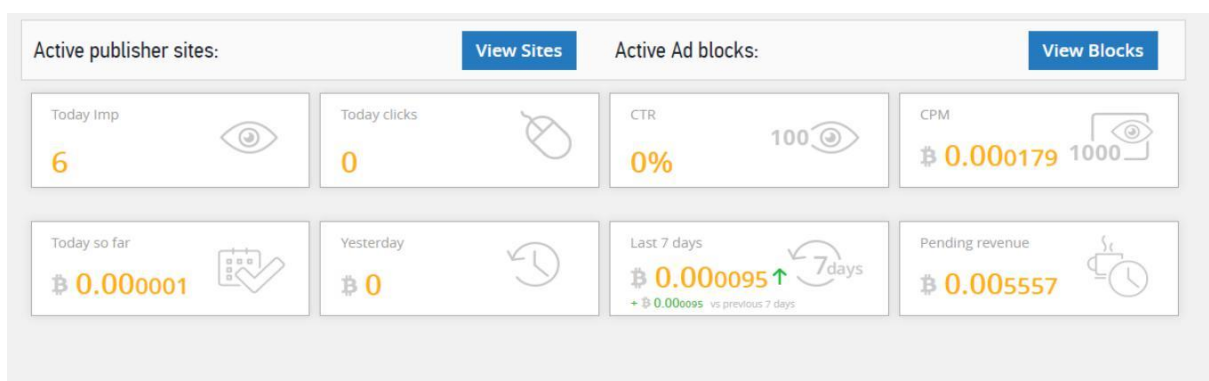


Рис. 4.4. Загальна статистика Видавця

- загальний дохід за останню добу;
- загальний дохід за сьогоднішній день;
- сумарна вартість наданих послуг за останні сім днів та відсоткові зміни в порівнянні з попереднім тижнем;
- очікуваний дохід.

Нижче розташовано статистичний блок з графічною інформацією.

Налаштування вибірки відбувається за наступними параметрами:

- початок періоду вибірки;
- кінцева дата періоду статистичної інформації;
- тип вибраних даних (покази або переходи за посиланнями рекламного матеріалу).

Скріншот статистичного рекламного блоку Видавця представлено на рис. 4.5.

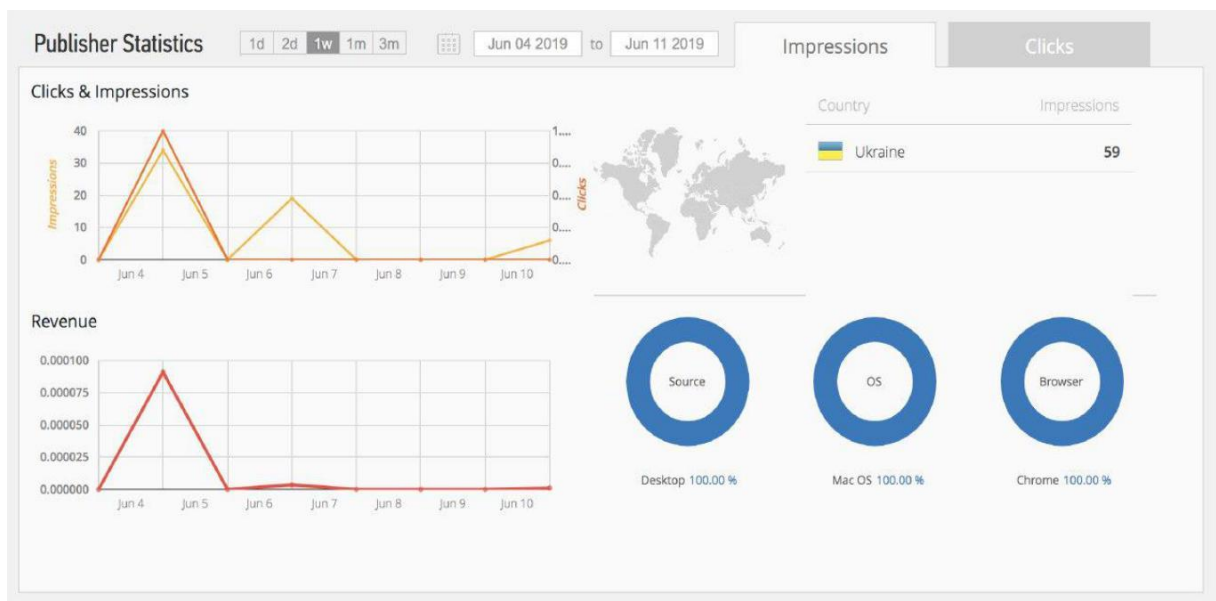


Рис. 4.5. Статистичний блок видавця за обраний період

Налаштування таргетингу здійснюється на сторінці кампанії за окремим посиланням. Скріншоти містяться на рис. 4.6 – 4.8. Вони включають наступні показники:

1. Лінгвістичні параметри – спеціальний селектор дозволяє вибрати мову кампанії.
2. Геолокація – таргетинг за регіоном перебування потенційного ліда. Доступні регіони: Сполучені Штати Америки та Канада, Центральна та Південна Америка, Європа, країни колишнього Радянського Союзу, Західна та Південна Азія, Австралія та Нова Зеландія, Африка, Океанія. Також доступна опція пріоритетних країн та членів чорного списку. Регіони внесені до першого частіше демонструватимуть маркетингові матеріали, що входять до складу кампанії, а другі ніколи не відображатимуть рекламних банерів, які буде створено в межах даної групи.

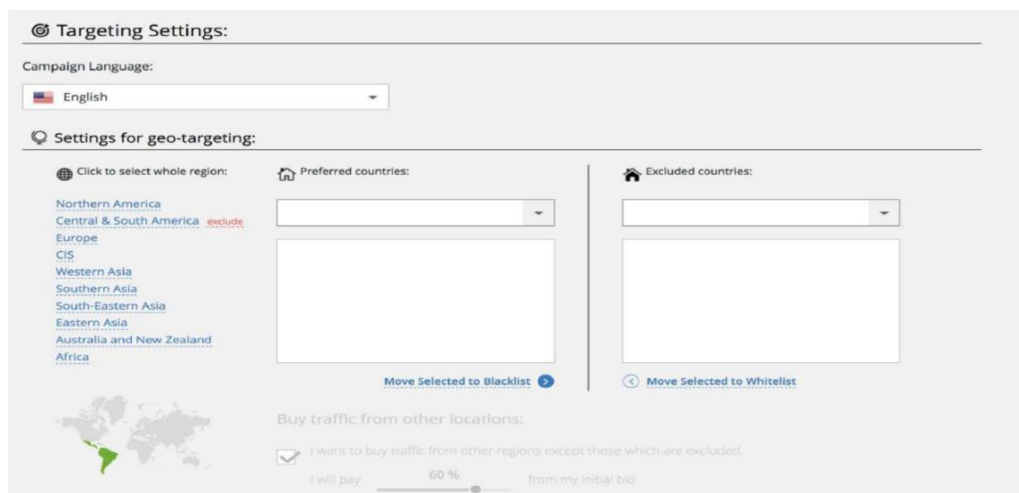


Рис. 4.6. Налаштування таргетингу за геолокаційним принципом

3. Пристрої – таргетинг за принципом платформи, на якій проводить маніпуляції користувач. Розділяється на мобільні та персональні комп'ютерні системи.
4. Додаткові опції включають в себе налаштування унікальних показів реклами, час для демонстрації маркетингового матеріалу та цикл скидання цих налаштувань на ініційовані.

5. Чорні та білі листи рекламних блоків. Дозволяють вибрати конкретні сайти для проведення кампанії за їхнім унікальним текстовим ідентифікатором. Також можна вилучати рекламні блоки з програми маркетингових заходів за тим самим принципом.

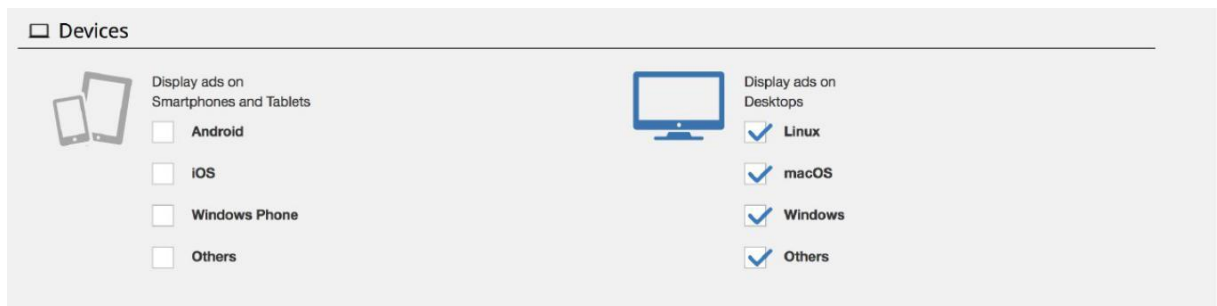


Рис. 4.7. Налаштування таргетингу за принципом операційної системи

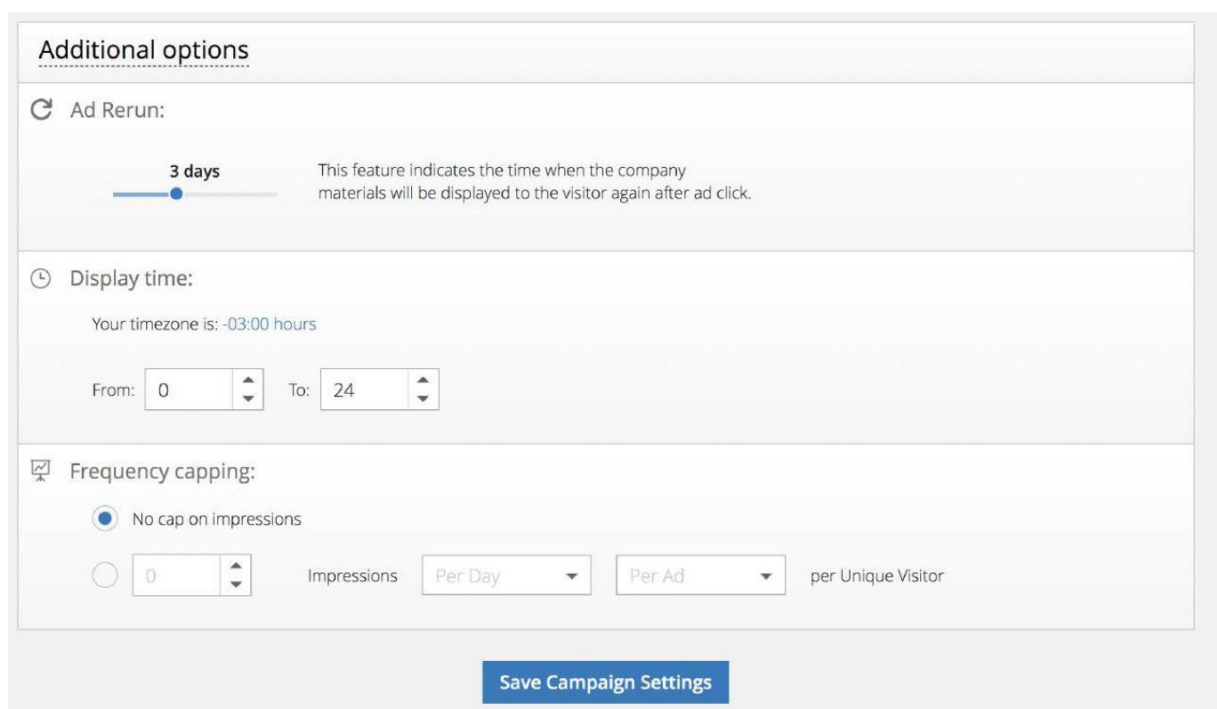


Рис. 4.8. Налаштування додаткових опцій таргетингу

Сторінка балансу містить кількість доступних на акаунті коштів у біткоїні і дозволяє поповнити рахунок для подальшого фінансування

рекламних кампаній користувача розробленого веб-застосунку. Для цього присутні наступні елементи:

- адреса біткоїн-гаманця для поповнення рахунку – її можна використати для транзакції своєї криптовалюти;
- qr-код можна використати у спеціалізованих додатках для автоматичного списання коштів.

Поповнення кампанії відбувається за рахунок фінансів, доступних на рахунку користувача. Скріншот форми наведено на рис. 4.9. Сторінка містить наступні елементи:

- коротку зведену інформацію про рекламну кампанію, а саме її назву, кошти, доступні для витрачення в рамках даного маркетингового заходу та приблизний аналог цієї суми у доларах;
- інформацію про загальний баланс на рахунку користувача;
- поле вводу для суми поповнення;
- кнопку підтвердження транзакції.

.123TEST Campaign Balance: ₿ 0 ~ \$ 0.00

Please Select Currency:

₿ BTC

Please Select Source of Payment:

Bitcoin Transaction Bitmedia Main Balance Other Ad Campaign

Available Balance

Balance in BTC

₿ 40.713395 ~ \$ 323,923.33

Send bitcoins from your main balance or split them between your ad campaigns.

Fund bonus list:

₿ 0.1+ 3% ₿ 0.5+ 5% ₿ 1+ 10%

Fund amount

₿ Please, enter the Bitcoin amount ~ \$ 0.00

FUND CAMPAIGN

Рис. 4.9. Робочий потік поповнення балансу кампанії

Користувач отримує статус Видавця при створенні першого власного сайту, після чого він отримає можливість створювати та керувати розміщеними там рекламними блоками. Для цього контролеру необхідно надати наступну текстову інформацію:

- url-адреса сайту користувача;
- вибір тематики сайту з-поміж запропонованих категорій (біткоїн, ігри, знайомства, фінанси, шопінг, техніка, медичні препарати, інше);
- мова веб-додатку;
- загальний опис застосунку.

4.2. Тестування веб-додатку

Контроль за якістю є невід'ємною частиною життєвого циклу розроблення програмного забезпечення являється частиною кожного етапу створення веб-додатку, як завершальна частина перед затвердженням імплементації функціональності.

Тестування направлено на виявлення помилок наступних типів:

- інтерфейсних (помилки стилізації, відображення сторінок);
- функціональних (перевірки правильності роботи частин додатку та інформаційних потоків);
- авторизаційних (захищеність від доступу до окремих елементів інтерфейсу та конфіденційних даних для незареєстрованих або з недостатнім рівнем доступу користувачів).

Одним із способів тестування ПЗ було ручне тестування, за допомогою якого вводилися різні дані до текстових полів, зокрема некоректні дані, пусті значення тощо. За результатами цього тестування було додано повідомлення для користувачів про відповідні помилки, які виникають при введенні ними некоректних даних.

Також для проведення тестування було використано Selenium. Це інструмент для автоматизованого управління веб-браузерами. Найбільш 45

популярною областю застосування Selenium є автоматизація тестування веб-додатків.

За допомогою тестування Selenium було отримані важливі результати щодо вдосконалення та доопрацювання клієнтської частини рекламної веб-платформи.

4.3. Порівняння розробки з існуючими аналогами

Розроблений веб-додаток відповідає усім визначеним вимогам, а саме:

- широкі можливості з налаштування таргетингу маркетингових компаній;
- гнучкі налаштування формату реклами – можливість обрати текст, анімацію, відео різних розмірів;
- гнучка тарифікація – можливість вибрати різні режими оплати, зокрема за кількість показів реклами або кількість переходів за посиланням;
- можливість перегляду детальної статистики по рекламним кампаніям та окремим банерам.

Окрім цього було усунуто наступні недоліки альтернативних програмних засобів:

- можливість оплати послуг рекламної платформи за допомогою біткоїн-гаманця;
- ширші можливості для контентної політики;
- можливість залишити заявку на те, щоб стати Видавцем (тобто право надавати рекламні місця на своєму сайті для демонстрації маркетингових матеріалів платформи).

Провівши аналіз розробленого програмного засобу, можна стверджувати, що він є достойним конкурентом для інших додатків, які присвячені наданню сучасних інформаційних онлайн маркетингових послуг.

4.4. Рекомендації щодо подальшого вдосконалення

Розроблений веб-застосунок можна застосовувати в сучасній комерційній маркетинговій діяльності, оскільки додаток надає достатньо функціональних потужностей. Проте сучасні бізнес індустрії дуже стрімко розвиваються і наданих можливостей може недовзі бути недостатньо.

Отже, розроблене програмне забезпечення можна і потрібно вдосконалювати, розширюючи його функціональність для подальшого використання у комерційних компаніях, що прагнуть просувати свої товари та послуги, а також у тих підприємців, що мають намір монетизувати свої веб-сайти для збільшення власних прибутків. Для цього можна надати наступні можливості:

- додавання операцій поповнення у традиційних валютах, зокрема долар, євро;
- інтеграція із сучасними платіжними системами, а саме Payoneer, Qiwi-гаманець;
- додавання рекламних банерів інтерактивного формату, а саме міні-ігри на основі canvas-моделі інтернет-браузера користувача;
- розширення статистичних метрик для графічного зображення, наприклад, конверсію;
- інтеграція з CRM-системами компаній;
- надання звітів за визначені календарні періоди для обраних рекламних кампаній у форматах pdf та csv з детальною статистикою кампаній та графічним зображенням цих даних;
- додання системи сповіщень електронною поштою для користувачів на різноманітні події, наприклад, різкі зміни в ефективності проведення кампанії, низький рівень фінансових рахунків профілю користувача сервісу тощо;
- перегляд загальної статистики ефективності роботи платформи за регіонами та категоріями для кращого розуміння необхідних налаштувань таргетингу;

- прив'язування банківських карток до сервісу для Рекламодавців та Видавців;
- розробка альтернативних версій додатку для різноманітних мобільних платформ;
- аналіз рекламних матеріалів користувачів для забезпечення передналаштувань таргетингу;
- авторизація через сторонні сервіси, наприклад Facebook, Google;
- режими спільного доступу користувачів до керування кампаніями для випадків, коли над ними працюють команди з декількох працівників-маркетологів.

ВИСНОВКИ

Метою дипломного проекту була розробка клієнтської частини рекламної веб-платформи, яка надавала б можливість оплати своїх послуг біткоіном та забезпечувала свободу контентної політики своїм користувачам.

Після проведення аналізу існуючих програмних аналогів, було зроблено висновок, що система має бути розроблена у вигляді веб-застосунку, що повинен мати зручний та зрозумілий інтерфейс, який мінімізуватиме помилки користувачів, та надаватиме надійну широку функціональність рекламодавцям та видавцям.

За результатами проведеного аналізу засобів реалізації, а саме мов програмування, бібліотек та фреймворків було прийнято рішення розробити клієнтську частину рекламної веб-платформи мовою JavaScript за допомогою фреймворку AngularJS за архітектурою шаблону MVC.

Розроблений додаток надає:

- можливість реєструватися і входити на сайт, як рекламодавець;
- можливість здійснювати оплату послуг біткоіном;
- можливість додавати власний сайт для розміщення там рекламних блоків інших користувачів;
- користувацький інтерфейс, спроектований за сучасними стандартами UI та UX.

Розробку програмного забезпечення виконано у повному обсязі та у відповідності до сформованих вимог. Тестування веб-застосунку виконано за затвердженою програмою та методикою тестування.

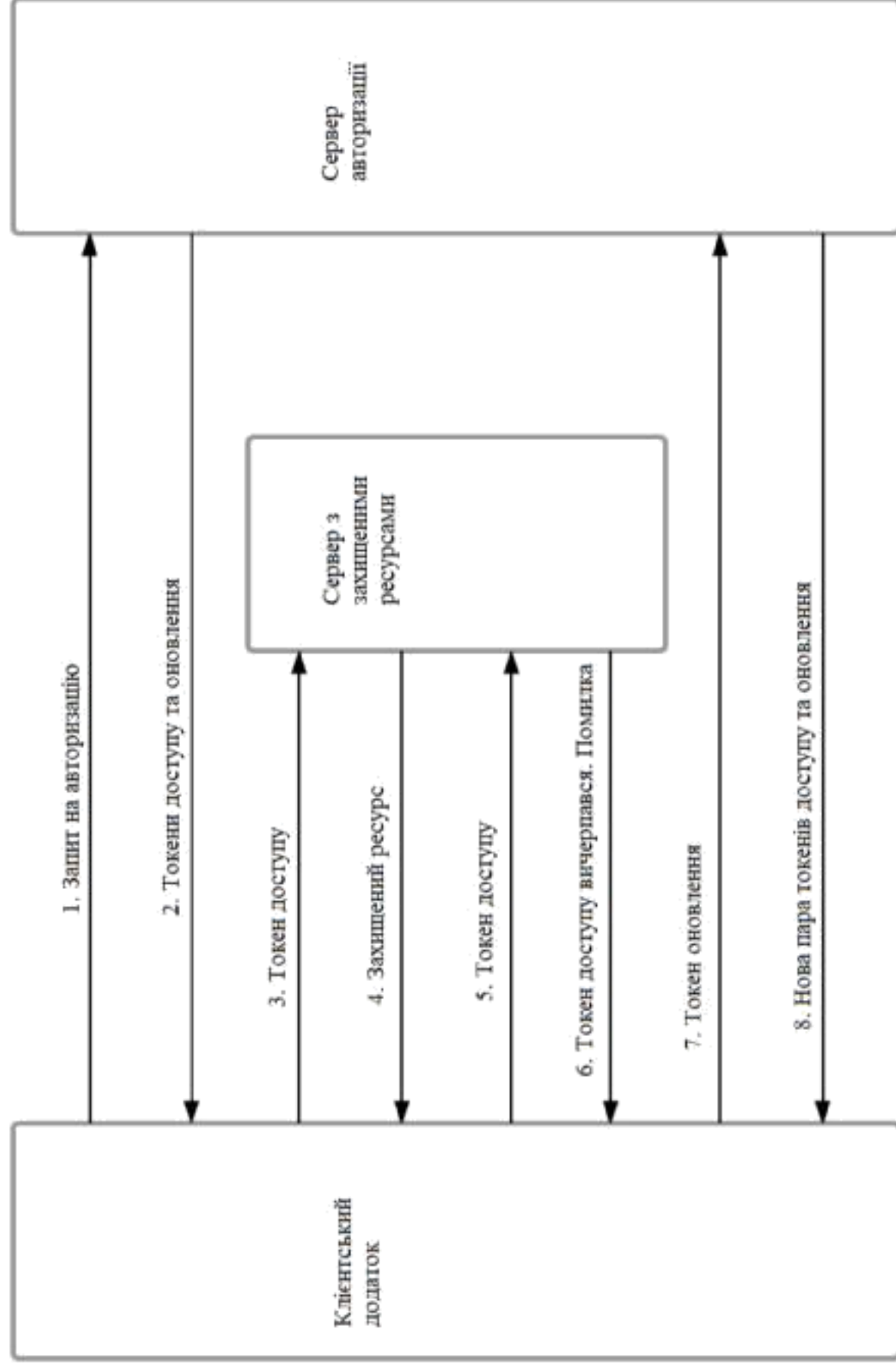
Використання розробленого веб-додатку забезпечить користувачам комфортний та зручний рекламний сервіс, що є простим у керуванні та доступним за різними типами тарифікації.

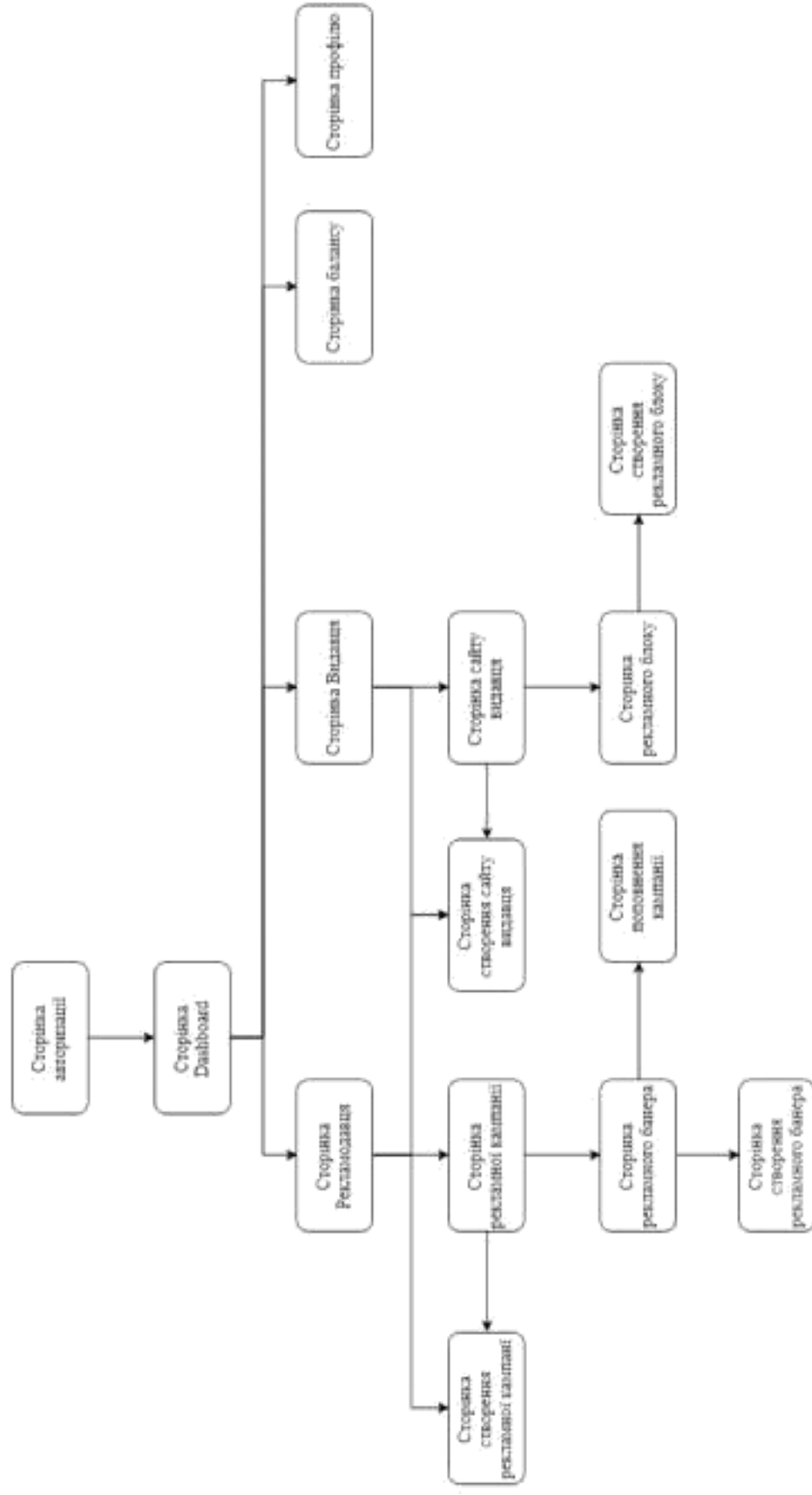
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Розмістіть рекламу в Google. [Електронний ресурс]. – дата візиту 12.04.2019. – Режим доступу до ресурсу: <https://ads.google.com/>
2. Google AdWords: Pros and Cons. [Електронний ресурс]. – дата візиту 13.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2sj2grM>
3. Facebook Ads Manager. [Електронний ресурс]. – дата візиту 13.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2DhTxdD>
4. Web application. [Електронний ресурс]. – дата візиту 20.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2Rm14yJ>
5. JavaScript. [Електронний ресурс]. – дата візиту 20.04.2019. – Режим доступу до ресурсу: <https://bit.ly/1RgM3Gw>
6. Современные возможности ES-2015. [Електронний ресурс]. – дата візиту 20.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2MJ3cBD>
7. What is PHP? [Електронний ресурс]. – дата візиту 25.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2vsqOz4>
8. Фреймворки в веб-разработке. [Електронний ресурс]. – дата візиту 25.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2KnO0Iv>
9. React (JavaScript library). [Електронний ресурс]. – дата візиту 30.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2t8Bylp>
10. Structuring React Projects—a Definitive Guide. [Електронний ресурс]. – дата візиту 30.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2Lw1uPV>
11. AngularJS. [Електронний ресурс]. – дата візиту 30.04.2019. – Режим доступу до ресурсу: <https://bit.ly/23HSWJf>
12. Pros and Cons of Vue.js Framework. [Електронний ресурс]. – дата візиту 30.04.2019. – Режим доступу до ресурсу: <https://bit.ly/2RgK4d8>
13. MVC Framework – Introduction. [Електронний ресурс]. – дата візиту 01.05.2019. – Режим доступу до ресурсу: <https://bit.ly/2kx0VaU>

ДОДАТКИ

Додаток 1
Копії графічних матеріалів

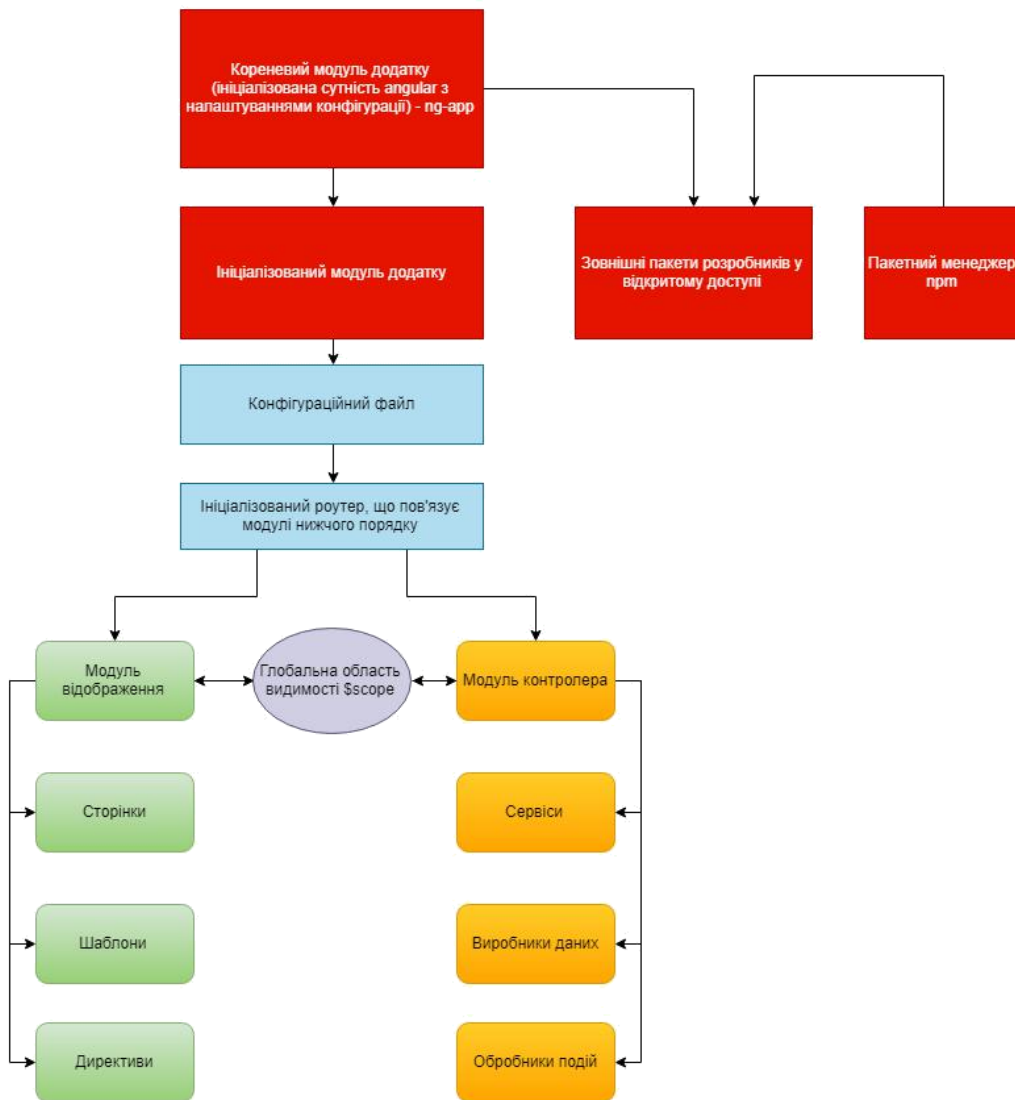




ДП. 045440-07-99

Веб-платформа для розміщення реклами. Клієнтська частина.

Граф переходів додатку. Граф переходів



Архітектура розробленого веб-додатку
Сорочинський В. В., група КП-52

Додаток 2
Копія презентації



Веб-платформа для розміщення реклами. Клієнтська частина

Виконав: ст. групи КП-52 Сорочинський В. В.

Науковий керівник: ст. к.т.н. Рибачок Н. А.

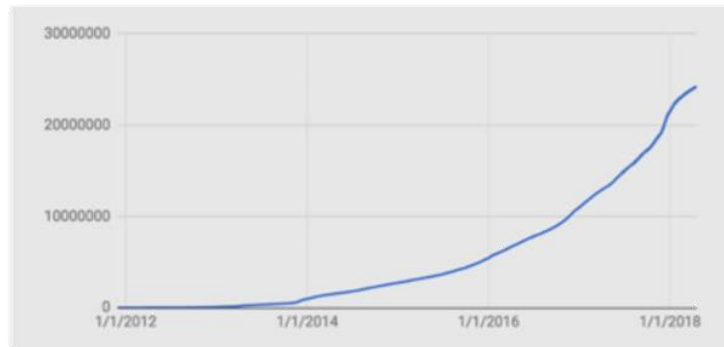
Київ – 2019

ПОСТАНОВКА ЗАДАЧІ



- **Мета проекту:** Розроблення клієнтської частини рекламної веб-платформи, яка дозволяє створювати, керувати та оплачувати за допомогою біткоіну маркетингові кампанії та рекламні банери.

АКТУАЛЬНІСТЬ



Зростання кількості адрес біткоїн-гаманців

3

Аналіз існуючих рішень

Google AdWords



Переваги:

- широкий інформаційний простір;
- детальна статистика за маркетинговими кампаніями;
- гнучкі налаштування таргетингу.

Facebook Ads Manager



Недоліки:

- жорстка контентна політика;
- обмежені опції для тарифікації послуг сервісів;
- недоступна оплата послуг біткоіном.

4



Основні характеристики системи

- Реєстрація та авторизація користувачів;
- можливість фінансування кампанії за допомогою біткоїну;
- гнучкі налаштування таргетингу;
- детальна статистика;
- керування рекламними кампаніями в реальному часі;
- різні формати рекламного банера.

5



Використані технології

- ✓ HTML & CSS
- ✓ JavaScript
- ✓ AngularJS
- ✓ Node.js



6



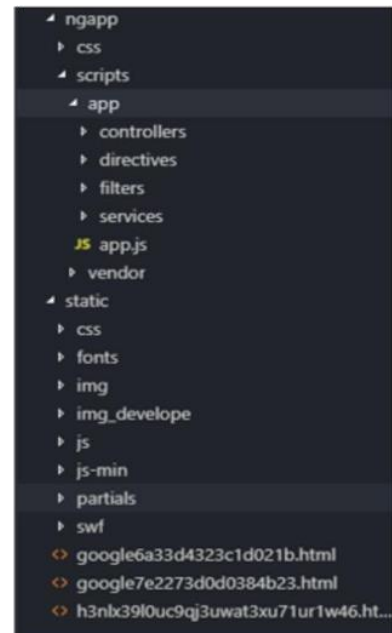
Архітектура проекту

Типи файлів:

- статичні;
- динамічні.

Перелік основних модулів:

- модуль авторизації;
- модуль рекламної кампанії;
- модуль балансу;
- модуль статистики;
- модуль рекламного блоку.



7



Авторизація

Типи користувачів:

- Рекламодавець (основний)
- Видавець (розширений)

8



Опис функціональності

Рекламодавець:

- створення та керування рекламними кампаніями;
- поповнення рахунку за допомогою біткоїну;
- створення та редагування рекламних банерів;
- перегляд статистики за рекламними кампаніями та банерами.

Видавець:

- додавання до платформи власних сайтів;
- створення та керування рекламними блоками.

9



Екран створення рекламної кампанії

[Dashboard](#) [Advertiser](#) [Publisher](#) [My balance](#) **₴40,713.94** **\$72,445.00** [PROFILE](#)

[Home](#) > [Ad Campaigns](#) > [Add New Campaign](#) ✓ Active Publisher (oleg+pu2@plascam.com)

Create Campaign


Create Adgroup

Set Targeting


TYPE OF CAMPAIGN

Choose the type of your advertising campaign according to your goals and requirements.

Please select your payment type:

**CPC**
COST PER CLICK

You choose the bid price for 1 click on your ad and pay only for clicks you get. This type of ads usually have a lower priority in rotation.

**CPM**
COST PER 1000 IMPRESSIONS

You choose the price per 1000 impressions for the ads of your ad campaign and pay for impressions. Such advertisements usually have a high priority in rotation.

10

Екран зведеної статистики рекламних кампаній



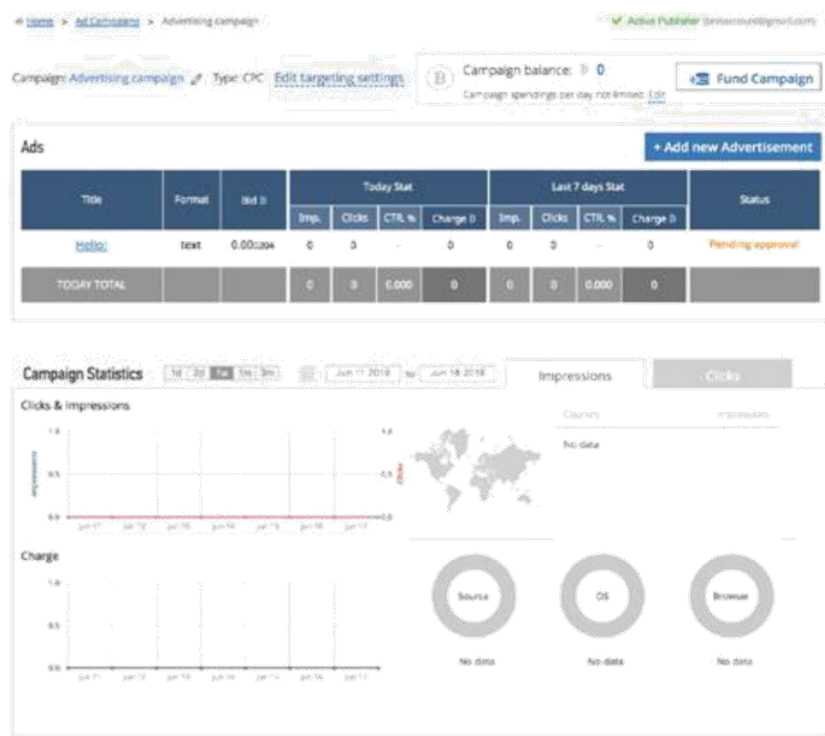
Home > Ad Campaigns Active Publisher @testaccount@gmail.com

Campaigns list + Add New Campaign

| Campaigns | Ads | Avg. Bid \$ | Type | Today Stat | | Last 7 days Stat | | Status | Balance \$ |
|--------------------------------------|-----|-------------|------|------------|--------|------------------|-----------|---------|---------------------|
| | | | | Imp. | Clicks | CTR, % | Charge \$ | | |
| Advertising campaign | 1 | 0.000204 | CPC | 0 | 0 | - | 0 | Stopped | 0 FUND |
| TOTAL | 1 | - | - | 0 | 0 | 0.000 | 0 | - | 0 |

11

Екран детальної статистики рекламних кампаній



12



Екран статусу рекламного банера

Home > Ad Campaigns > Advertising campaign > Hello!

Active Publisher (testaccount@gmail.com)

Today

0 Clicks

0

Last 7 days

0 Clicks

0

Last 28 days

0 Clicks

0

Ad Preview

Modify Ad

Hello!

perfect day

perfect light

google.com

>

| Status | Click bid |
|--------------|----------------------------|
| UNDER REVIEW | Click bid: 0.0002046 ✓ Set |

13



Висновки

- Розроблено архітектуру та кодову базу веб-платформи для розміщення реклами;
- враховано головні недоліки аналогів;
- сформована функціональність для подальшого вдосконалення проекту.

14

Перевірка на антиплагіат

| Розділ | Процент унікальності |
|--------|----------------------|
| 1 | 98% |
| 2 | 97% |
| 3 | 99% |
| 4 | 99% |
| Диплом | 96% |

15



Дякую за увагу!

Додаток 3
Лістинг програми

```

angular.module('AdvertisementApp')
  .controller('AdAddAdvertPageCtrl', ['$rootScope', '$scope', '$http',
'$routeParams', '$location', 'logger', 'errorReporter', '$q', '$timeout',
'$uibModal', 'md5', function ($rootScope,

$scope,

$http,

$routeParams,

$location,

logger,

errorReporter, $q, $timeout, $uibModal, md5) {

  $scope.campaignId = $routeParams.campaignId;
  $scope.campaignEpmId = -1;

  if ( $rootScope.adCampaigns ) {
    $scope.campaignName = $rootScope.adCampaigns.filter(function (c)
{return c.campaignId === $scope.campaignId;})[0].campaignName;
  } else {
    $scope.campaignName = 'Campaign';
  }

  $scope.submitBtnText = 'Save Advertisement';
  $scope.acceptedAdDimensions = [];
  $scope.inputErr = false;
  //$scope.videoIsLoading = false;
  $scope.isAdTypeText = false;
  $scope.isAdTypeImage = false;
  $scope.isAdTypeVideo = false;
  $scope.isAdTypeResponsive = false;
  $scope.filesToSend = [];
  $scope.videoToSend = '';
  $scope.rawBackgroundResponsiveAdImage = '';
  $scope.rawLogoResponsiveAdImage = '';
  $scope.backgroundResponsiveAdImage = '';
  $scope.logoResponsiveAdImage = '';
  $scope.backgroundResponsiveAdImageUrl = '';
  $scope.logoResponsiveAdImageUrl = '';

  $scope.campaign_cpm = false;
  $scope.campaign_cpc = false;
  $scope.campaign_video = false;

  $scope.videoCheckSizeError = false;
  $scope.responsiveCheckLogoDimensions = false;
  $scope.responsiveCheckBackgroundDimensions = false;
  $scope.errorTipVideoMessage = 'Maximum video size is 20Mb, available format -
mp4';
  $scope.errorTipResponsiveMessage = 'Maximum image size is 200KB';

```

```

$scope.errorTipResponsiveMessageLogoSize = 'Logo must be 128x128px';
$scope.errorTipResponsiveMessageBackgroundSize = 'Background must be
336x178px';

var maxVideoSize = 20000000;
var maxResponsiveAdSize = 204800; //200KB

var gaussianFunction = function (x, sigma) {
    var SIGMA = (sigma) ? $scope.bidParams.SIGMA + 0.6 : $scope.bidParams.SIGMA
+ 0.4;
    return (1 / (SIGMA * Math.sqrt(2 * Math.PI))) * Math.exp(-1 * (Math.pow(x -
$scope.bidParams.M, 2) / (2 * Math.pow(SIGMA, 2))));
};

var normalizeCurrentBid = function (bid) {
    var result = (parseFloat(bid) - $scope.bidParams.minBid) /
($scope.bidParams.maxBid - $scope.bidParams.minBid);
    return result;
};

var getAdRank = function (bid) {
    if ( $scope.bidParams ) {

        var adRank;
        if ($scope.campaign_cpc) {
            adRank = gaussianFunction(normalizeCurrentBid(bid), false) * 100
/ ($scope.bidParams.gaussianToPercentRatio - 0.42);
        } else if ($scope.campaign_cpm){
            adRank = gaussianFunction(normalizeCurrentBid(bid), true) * 100
/ ($scope.bidParams.gaussianToPercentRatio - 0.511);
        }

        var red = parseInt(225 * (1 - (adRank / 100)));
        red = red < 0 ? 0 : red > 255 ? 255 : red;
        red = red.toString(16);
        if ( red.length === 1 ) {
            red = '00';
        }

        var green = parseInt(195 * (adRank / 100));
        green = green > 255 ? 255 : green; green =
green.toString(16);
        if ( green.length === 1 ) {
            green = '00';
        }

        var blue = parseInt(225 * gaussianFunction((adRank / 100) +
0.2)); blue = blue > 255 ? 255 : blue < 0 ? 0 : blue; blue =
blue.toString(16);
        if ( blue.length === 1 ) {
            blue = '00';
        }

        var rankColor = '#' + red + green + blue;
    }
}

```

```

var rankText;

if ( adRank < 30 ) {
    rankText = 'Poor visibility';
} else if ( adRank >= 30 && adRank < 50 ) {
    rankText = 'Low visibility';
} else if ( adRank >= 50 && adRank < 60 ) {
    rankText = 'Normal visibility';
} else if ( adRank >= 60 && adRank < 70 ) {
    rankText = 'Good visibility';
} else if ( adRank >= 70 && adRank < 81 ) {
    rankText = 'Nice visibility';
} else if ( adRank >= 81 && adRank < 91 ) {
    rankText = 'On top of visible ads';
} else {
    rankText = 'Maximum visibility';
}

return {
    adRank:    adRank,
    rankColor: rankColor,
    rankText:  rankText
};
}
};

var loadBidParams = function () {
    $q.all({
        bid: $http.get('/ngapp/data/ad/bidParams/' + $scope.campaignId),
        type: $http.get('/ngapp/data/campaign/type/' +
            $scope.campaignId) }).then(function(res){
            // get campaign type
            $scope.campaignType = res.type.data.type;
            if ($scope.campaignType == 'native_cpc'){
                $scope.campaign_cpc = true;
                $scope.campaign_cpm = false;
                $scope.campaign_video = false;
            } else if($scope.campaignType == 'native_cpm'){
                $scope.campaign_cpm = true;
                $scope.campaign_cpc = false;
                $scope.campaign_video = false;
            } else if($scope.campaignType == 'video' || $scope.campaignType
== 'epom'){
                $scope.campaign_cpm = false;
                $scope.campaign_cpc = false;
                $scope.campaign_video = true;
            }

            $scope.bidParams = res.bid.data;
            $scope.bidStep = (res.bid.data.maxBid - res.bid.data.minBid) / 100;
            $scope.imgAdClickBid.maxPrice = res.bid.data.maxBid;
            $scope.imgAdClickBid.minPrice = res.bid.data.minBid;
            $scope.newAd.clickPrice.maxPrice = res.bid.data.maxBid;

```



```

    $scope.newAd.clickPrice.minPrice = res.bid.data.minBid;
    var suggestedBid = (((res.bid.data.maxBid - res.bid.data.minBid) *
0.63) + res.bid.data.minBid).toFixed(7);
    $scope.imgAdClickBid.numberStr = String(suggestedBid);
    $scope.newAd.clickPrice.numberStr = String(suggestedBid);
    $scope.imageAdRank = getAdRank($scope.imgAdClickBid.numberStr);
    $scope.textAdRank = getAdRank($scope.newAd.clickPrice.numberStr);
  });
};

```

```

$scope.selectAdType = function (status) {
  switch (status){
    case 'text':
      $scope.isAdTypeText = true;
      $scope.isAdTypeImage = false;
      $scope.isAdTypeVideo = false;
      $scope.isAdTypeResponsive = false;
      break;
    case 'image':
      $scope.isAdTypeText = false;
      $scope.isAdTypeImage = true;
      $scope.isAdTypeVideo = false;
      $scope.isAdTypeResponsive = false;
      break;
    case 'video':
      $scope.isAdTypeText = false;
      $scope.isAdTypeImage = false;
      $scope.isAdTypeVideo = true;
      $scope.isAdTypeResponsive = false;
      break;
    case 'responsive':
      $scope.isAdTypeText = false;
      $scope.isAdTypeImage = false;
      $scope.isAdTypeVideo = false;
      $scope.isAdTypeResponsive = true;
      break;
  }
};

```

```

// load campaign type CPC , CPM
$http.get('/ngapp/data/campaign/ad/' + $scope.campaignId)
.success(function (res) {
  $scope.campaignType = res.type ;
  if ( res.type == 'native_cpc' ) {
    $scope.campaign_cpc = true;
    $scope.campaign_cpm = false;
    $scope.campaign_video = false;
    $scope.selectAdType('responsive');
  } else if ( res.type == 'native_cpm' )
  { $scope.campaign_cpm = true;
    $scope.campaign_cpc = false;
    $scope.campaign_video = false;
    $scope.selectAdType('responsive');
  } else if ( res.type == 'video' || res.type == 'epom' ) {

```

```

        $scope.campaign_cpm = false;
        $scope.campaign_cpc = false;
        $scope.campaign_video = true;
        $scope.selectAdType('video');
    }
})
.error(function (err) {
    console.log(err);
});

$http.get('/ngapp/data/campaign/epomId/' + $scope.campaignId)
    .success(function (res) {
        $scope.campaignEpomId = res.epomId;
    })
    .error(function (err) {
        console.log(err);
    });

$http.get('/ngapp/data/ad/addadvert/available_dimensions')
    .success(function (data) {
        $scope.acceptedAdDimensions = data.sizes;
    });

var getCookie = function (cname) {
    var name = cname + '=';
    var ca = document.cookie.split(';');
    for ( var i = 0; i < ca.length; i++ ) {
        var c = ca[i];
        while (c.charAt(0) == ' ') {
            c = c.substring(1);
        }
        if ( c.indexOf(name) === 0 ) {
            return c.substring(name.length, c.length);
        }
    }
    return '';
};

$scope.removeImg = function (index) {
    $scope.filesToSend.splice(index, 1);
};

$scope.isDimensionAdded = function (dimension)
{
    var filter = function (el) {
        return el.dimensions === dimension;
    };

    return $scope.filesToSend.filter(filter).length;
};

$scope.submitImgAd = function () {

    if ( $scope.filesToSend.length ) {

```

```

if ( !$scope.imgAdTitle.inputOk ) {
    $scope.imgAdTitle.inputErr = true;
}

if ( !$scope.imgAdClickUrl.inputOk ) {
    $scope.imgAdClickUrl.inputErr = true;
    $scope.imgAdClickUrl.red_border = true;
}

if ( $scope.imgAdTitle.inputOk && $scope.imgAdClickUrl.inputOk )
{
    $scope.submitBtnText = 'Please wait';
    var formData = new FormData();
    for ( var i = 0; i < $scope.filesToSend.length; i++ ) {
        var name = $scope.filesToSend[i].dimensions + '.' +
$scope.filesToSend[i].name.split('.')
        .splice(-1);
        formData.append('file', $scope.filesToSend[i], name);
    }

    formData.append('campaignId', $scope.campaignId);
    formData.append('adType', 'image');
    formData.append('imageType', $scope.graphicsAdType);
    formData.append('language', $scope.imgLanguageObj.item);
    formData.append('clickUrl', ($scope.imgAdClickUrl.isSiteProtocolSecure
? 'https://' : 'http://') + $scope.imgAdClickUrl.urlStr);
    formData.append('clickPrice', $scope.imgAdClickBid.numberStr);
    formData.append('title', $scope.imgAdTitle.textStr);
    formData.append('campaignType', $scope.campaignType);

    var xhr = new XMLHttpRequest();
    xhr.open('POST', '/ngapp/data/ad/addadvert?imageType='
+ $scope.graphicsAdType);
    xhr.setRequestHeader('X-XSRF-TOKEN', getCookie('XSRF-TOKEN'));
    xhr.onerror = function () {
        logger.logError('New Image Advert', 'Network error, please check your
Internet connection');
    };

    xhr.onload = function () {
        if ( xhr.status == 200 ) {
            var newAdvertId = JSON.parse(xhr.response).advertId;
            logger.logSuccess('New Image Advert', '"' +
$scope.imgAdTitle.textStr + '" was added successfully');
            $location.path('/ad/item/' + newAdvertId + '/image/ ');
            $rootScope.$apply();
        } else {
            logger.logError('New Image Advert', 'Unexpected error occured,
please try again later');
            errorReporter.report('/ngapp/data/ad/addadvert?imageType=' +
$scope.graphicsAdType, formData, xhr.status, xhr.responseText);
        }
    };
}

```

```

        xhr.send(formData);
    } else { $scope.inputErr
        = true;
    }
}
};

$scope.$watchGroup(['rawBackgroundResponsiveAdImage',
'rawLogoResponsiveAdImage'], function () {
    $scope.responsiveCheckSizeError = ($scope.rawBackgroundResponsiveAdImage &&
    $scope.rawBackgroundResponsiveAdImage.size > maxResponsiveAdSize) ||
    ($scope.rawLogoResponsiveAdImage && $scope.rawLogoResponsiveAdImage.size >
    maxResponsiveAdSize);
    });

$scope.responsiveCreateAdButtonDisabled = function() {
    return !$scope.backgroundResponsiveAdImage || !$scope.logoResponsiveAdImage
        || !$scope.newResponsiveAd.headline.textStr
        || !$scope.newResponsiveAd.description.textStr ||
    !$scope.newResponsiveAd.businessName.textStr
        || !$scope.newResponsiveAd.clickUrl.urlStr ||
    !$scope.newResponsiveAd.clickUrl.inputOk || $scope.responsiveCheckSizeError ||
        $scope.responsiveCheckLogoDimensions ||
    $scope.responsiveCheckBackgroundDimensions;
    };

$scope.submitResponsiveAd = function () {
    if ( $scope.backgroundResponsiveAdImage && $scope.logoResponsiveAdImage &&
    !$scope.responsiveCheckSizeError && !$scope.responsiveCheckLogoDimensions &&
    !$scope.responsiveCheckBackgroundDimensions) {
        if ( !$scope.newResponsiveAd.headline.inputOk ) {
            $scope.newResponsiveAd.headline.inputErr = true;
        }
        if ( !$scope.newResponsiveAd.description.inputOk ) {
            $scope.newResponsiveAd.description.inputErr = true;
        }
        if ( !$scope.newResponsiveAd.businessName.inputOk ) {
            $scope.newResponsiveAd.businessName.inputErr = true;
        }
        if ( !$scope.newResponsiveAd.clickUrl.inputOk ) {
            $scope.newResponsiveAd.clickUrl.inputErr = true;
            $scope.newResponsiveAd.clickUrl.red_border = true;
        }

        if ( $scope.newResponsiveAd.clickUrl.inputOk &&
    $scope.newResponsiveAd.headline.inputOk &&
        $scope.newResponsiveAd.description.inputOk &&
    $scope.newResponsiveAd.businessName.inputOk) {
            $scope.submitBtnText = 'Please wait';
            var formData = new FormData();

            formData.append('logoFile',
    $scope.logoResponsiveAdImage, $scope.logoResponsiveAdImage.name);

```

```

        formData.append('backgroundFile',
$scope.backgroundResponsiveAdImage, $scope.backgroundResponsiveAdImage.name);
        formData.append('campaignId', $scope.campaignId);
        formData.append('adType', 'responsive');
        formData.append('headline', $scope.newResponsiveAd.headline.textStr);
        formData.append('description',
$scope.newResponsiveAd.description.textStr);
        formData.append('businessName',
$scope.newResponsiveAd.businessName.textStr);
        formData.append('language', $scope.imgLanguageObj.item);
        formData.append('clickUrl',
($scope.newResponsiveAd.clickUrl.isSiteProtocolSecure ? 'https://' : 'http://') +
$scope.newResponsiveAd.clickUrl.urlStr);
        formData.append('clickPrice', $scope.imgAdClickBid.numberStr);
        formData.append('title', $scope.newResponsiveAd.headline.textStr);
        formData.append('campaignType', $scope.campaignType);

var xhr = new XMLHttpRequest();
xhr.open('POST', '/ngapp/data/ad/addresponsiveadvertisemnet');
xhr.setRequestHeader('X-XSRF-TOKEN', getCookie('XSRF-TOKEN'));
xhr.onerror = function () {
    logger.logError('New Responsive Advert', 'Network error, please check
your Internet connection');
};

xhr.onload = function () {
    if ( xhr.status == 200 ) {
        var newAdvertId = JSON.parse(xhr.response).advertId;
        logger.logSuccess('New Responsive Advert', '"' +
$scope.newResponsiveAd.headline.textStr + '" was added successfully');
        $location.path('/ad/item/' + newAdvertId + '/responsive/ ');
        $rootScope.$apply();
    } else {
        logger.logError('New Responsive Advert', 'Unexpected error occurred,
please try again later');
        errorReporter.report('/ngapp/data/ad/addresponsiveadvertisemnet',
formData, xhr.status, xhr.responseText);
    }
};

xhr.send(formData);
} else {
    $scope.inputErr = true;
}
}
};

const getHash = function (stringToHash) {
    return crypto.createHash('md5').update(stringToHash).digest("hex");
};

$scope.videoCreateAdButtonDisabled = function() {
    return !$scope.videoToSend || !$scope.videoAdTitle.textStr ||
!$scope.videoAdClickUrl.urlStr || $scope.videoAdTitle.inputErr ||

```

```

$scope.videoAdClickUrl.inputErr || $scope.isSending ||
$scope.videoCheckSizeError;
    });

    $scope.$watch('videoToSend', function () {
        $scope.videoCheckSizeError = !($scope.videoToSend !== undefined &&
        $scope.videoToSend.size < maxVideoSize);
    });

    $scope.submitVideoAd = function () {
        if ( $scope.videoToSend ) {
            if ( !$scope.videoAdTitle.inputOk ) {
                $scope.videoAdTitle.inputErr = true;
            }

            if ( !$scope.videoAdClickUrl.inputOk ) {
                $scope.videoAdClickUrl.inputErr = true;
                $scope.videoAdClickUrl.red_border = true;
            }

            if ( $scope.videoAdTitle.inputOk && $scope.videoAdClickUrl.inputOk &&
            !$scope.isSending && !$scope.videoCheckSizeError) {
                $scope.submitBtnText = 'Please wait';
                $scope.isSending = true;

                $http.get('/ngapp/data/profile/epomusername')
                    .success(function (data, status) {
                        if(status === 200) {
                            //$scope.videoIsLoading = true;
                            const username = data.username;
                            const timestamp = data.timestamp;
                            const hash = data.hash;
                            var formData = new FormData();

                            formData.append('hash', hash);
                            formData.append('timestamp', timestamp);
                            formData.append('username', username);

                            formData.append('placementType', 'VIDEO_PLACEMENT');
                            formData.append('campaignId', $scope.campaignEpomId);
                            formData.append('active', false);
                            formData.append('name', $scope.videoAdTitle.textStr);
                            formData.append('weight', 1);
                            formData.append('bannerType', 'LOCAL_VIDEO');
                            formData.append('imageBannerLink',
                                ($scope.videoAdClickUrl.isSiteProtocolSecure ? 'https://' : 'http://') +
                                $scope.videoAdClickUrl.urlStr);
                            formData.append('url',
                                ($scope.videoAdClickUrl.isSiteProtocolSecure ? 'https://' : 'http://') +
                                $scope.videoAdClickUrl.urlStr);
                            formData.append('videoFile', $scope.videoToSend);
                            formData.append('inStreamVideoType', 'LINEAR_VIDEO_AD');

                            var config = {

```

```

        headers: {'Content-type': 'multipart/form-
data'} });

        axios.post('https://n1535.epom.com/rest-api/banner/create.do',
formData, config)
        .then(function (response) {
            if ( response.status === 200 ) {
                var newAdvertId = response.data.id;
                var newAdvert = response.data;
                $http.post('/ngapp/data/ad/setupvideoadvert',
{ epomVideoAdvertisement: newAdvert}).
                    success(function(data, status) {
                        if(status === 200) {
                            logger.logSuccess('New Video Advert', '"' +
$scope.videoAdTitle.textStr + '" was setuped successfully');
                            $scope.isSending = false;
                        }
                        else {
                            logger.logError('New Video Advert Setup', 'Unexpected
error occured, please try again later');

errorReporter.report('ngapp/data/ad/setupvideoadvert', { epomVideoAdvertisement:
newAdvert}, status, data);

                            $scope.isSending = false;
                        }
                    }).
                    error(function(data, status) {
                        logger.logError('New Video Advert Setup', 'Unexpected
error occured, please try again later');
                        errorReporter.report('ngapp/data/ad/setupvideoadvert',
{ epomVideoAdvertisement: newAdvert}, status, data);
                        $scope.isSending = false;
                    });

                logger.logSuccess('New Video Advert', '"' +
$scope.videoAdTitle.textStr + '" was added successfully');
                $scope.isSending = false;
                $location.path('/ad/item/' + newAdvertId + '/video/ ');
//todo: refactor to mongo_id
                $rootScope.$apply();
            } else {
                logger.logError('New Video Advert', 'Unexpected error
occured, please try again later');
                $scope.isSending = false;
                errorReporter.report('https://n1535.epom.com/rest-
api/banner/create.do', formData ,response.status, response.data);
            }
        })
        .catch(function(response) {
            logger.logError('New Video Advert', 'Something went
wrong, please try again later');
            $scope.isSending = false;
            errorReporter.report('https://n1535.epom.com/rest-
api/banner/create.do', formData ,response.status, response.data);

```

```

        })

        } else if (status === 403) {
            logger.logError('New Video Advert', 'No access to
video advertisements');
            $scope.isSending = false;
        }
    })
    .error(function (err) {
        $scope.isSending = false;
    });

    } else { $scope.inputErr
    = true;
    }
    }
};

```


Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

ВЕБ-ПЛАТФОРМА ДЛЯ РОЗМІЩЕННЯ РЕКЛАМИ. КЛІЄНТСЬКА
ЧАСТИНА

Програма та методика тестування

ДП.045440-04-51

«ПОГОДЖЕНО»

Керівник проекту:

_____ Н.А. Рибачок

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ В.В. Сорочинський

ЗМІСТ

| | |
|--------------------------------------|---|
| 1. Об'єкт випробувань..... | 3 |
| 2. Мета тестування..... | 3 |
| 3. Методи тестування..... | 3 |
| 4. Засоби та порядок тестування..... | 4 |
| 5. Висновки..... | 5 |

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Програмне забезпечення для клієнтської частини рекламної веб-платформи є AngularJS-додатком, написаний на мові програмування JavaScript (клієнтська частина) в межах архітектури Модель-Відображення-Контролер.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- 1) Взаємодія клієнта з сервером.
- 2) Забезпечення коректного відображення даних з усіх відомих веб-браузерів.
- 3) Забезпечення коректного відображення даних з усіх відомих платформ.
- 4) Забезпечення належного рівня безпеки даних.
- 5) Відповідність програмного забезпечення вимогам Технічного завдання.

3. МЕТОДИ ТЕСТУВАННЯ

Тестування виконується методом Gray Box Testing. Перевіряється як код, так і безпосередньо програмний продукт на відповідність функціональним вимогам. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- 1) Тестування базового функціоналу (smoke test).
- 2) Тестування продуктивності програмного забезпечення, особливо тестування стабільності (performance testing).
- 3) Навантажувальне тестування, особливо перевірка коректної роботи через мобільний інтернет (load testing).

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Працездатність веб-сервісу перевіряється шляхом:

- 1) динамічного ручного тестування – введенням граничних та недопустимих значень в поля, які можна редагувати;
- 2) динамічного ручного тестування на відповідність функціональним вимогам;
- 3) статичного тестування коду;
- 4) тестування сервісу в різних браузерях та на різних пристроях;
- 5) тестування при максимальному навантаженні;
- 6) тестування інтерфейсу.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“__” _____ 2019 р.

ВЕБ-ПЛАТФОРМА ДЛЯ РОЗМІЩЕННЯ РЕКЛАМИ. КЛІЄНТСЬКА
ЧАСТИНА

Керівництво користувача

ДП.045440-05-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Н.А. Рибачок

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ В.В. Сорочинський

ЗМІСТ

| | |
|--|---|
| 1. Опис структури додатку | 3 |
| 2. Опис вмісту веб-сторінок | 4 |
| 3. Процедура авторизації користувача | 5 |
| 4. Заявка на видавництво | 5 |

1. Опис структури додатку

Інформаційна система представлена односторінковим веб-додатком, який складається зі статичних і динамічних модулів, які взаємодіють із сервером. Основне призначення – створення, керування рекламними кампаніям, блоками, перегляд статистики та поповнення балансу кампаній за допомогою біткоїну.

Застосунок включає наступні модулі:

- «Login»;
- «Dashboard»;
- «Advertiser»;
- «Publisher»;
- «My balance»;
- «Profile»;
- «Become a publisher».

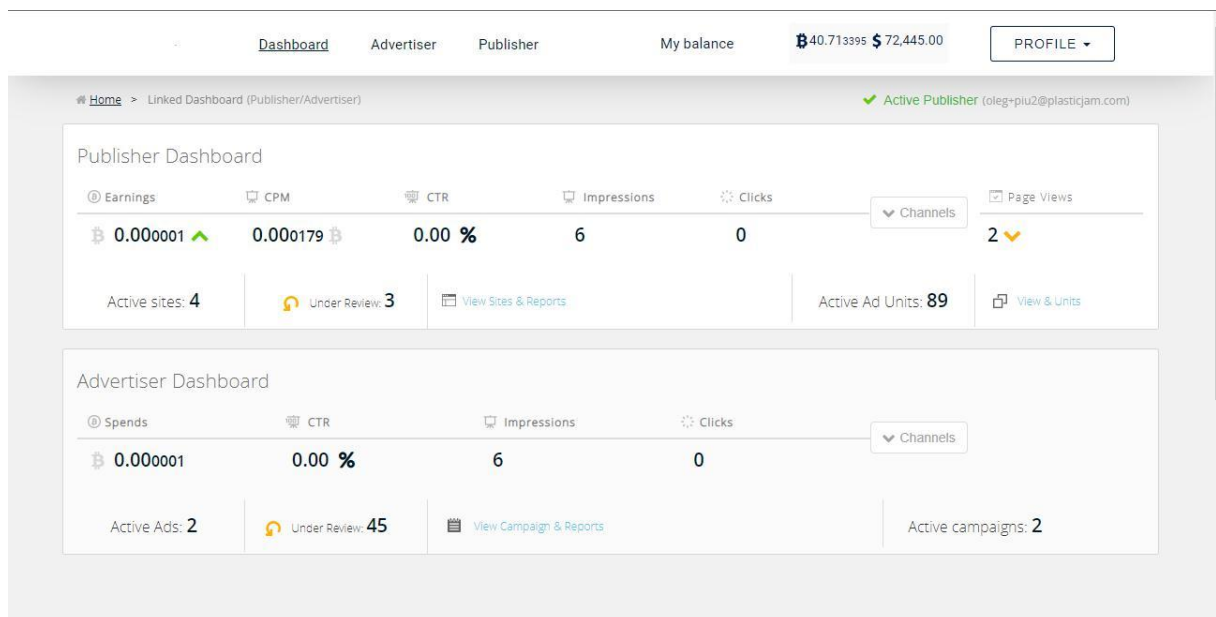
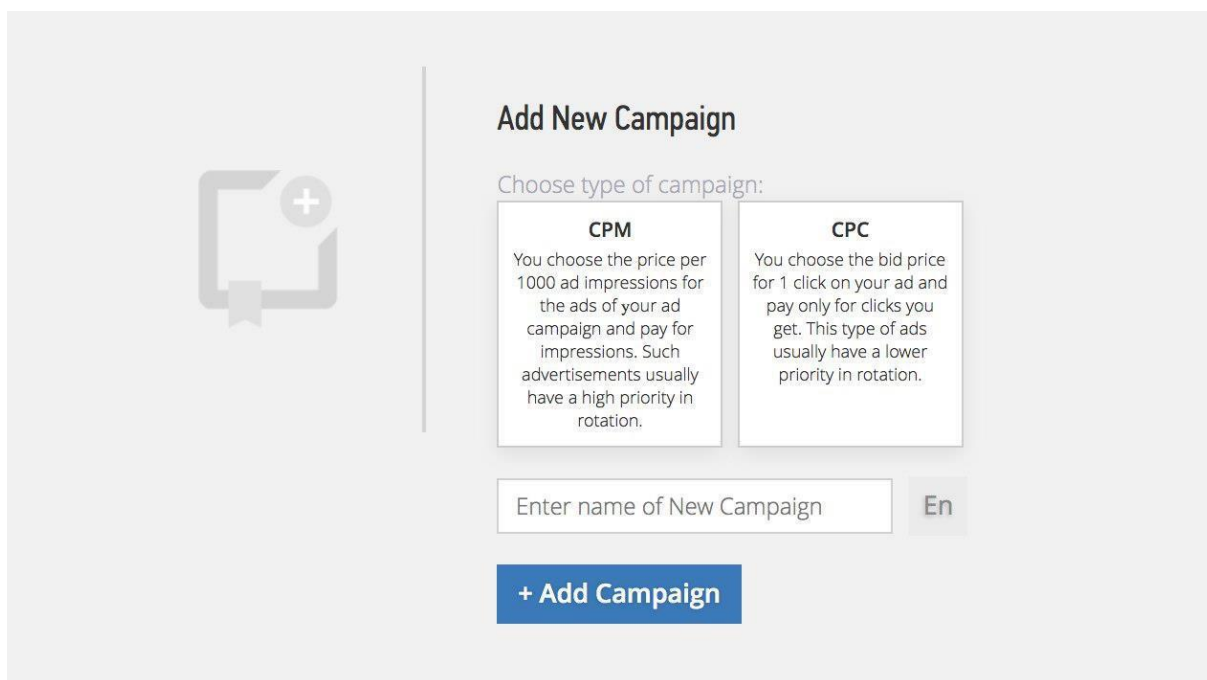


Рис. 1. «Dashboard сторінка»

У хедері веб-додатку містяться посилання для переходу в іншу вкладку.

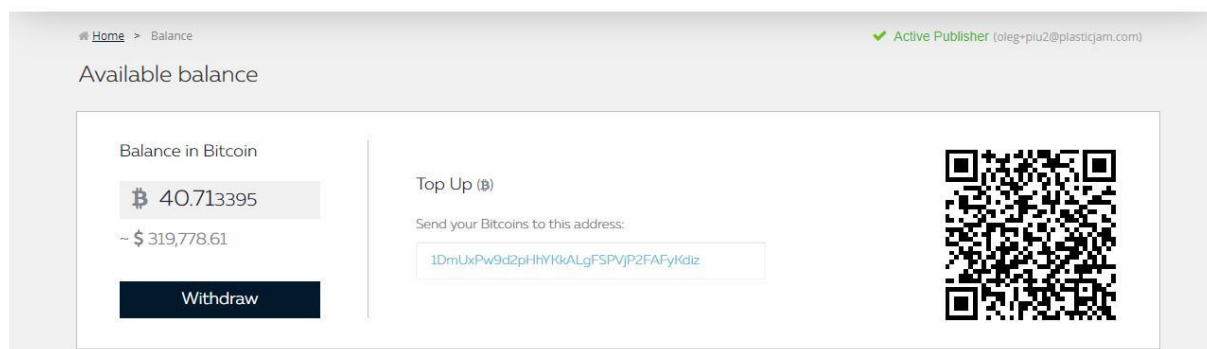


The screenshot shows a form titled "Add New Campaign". On the left is a large icon of a document with a plus sign and a bookmark. The form itself has a heading "Add New Campaign" and a sub-heading "Choose type of campaign:". Below this are two boxes: "CPM" and "CPC". The "CPM" box explains that you choose the price per 1000 ad impressions and that such advertisements usually have a high priority in rotation. The "CPC" box explains that you choose the bid price for 1 click and pay only for clicks you get, noting that this type of ads usually have a lower priority in rotation. Below these boxes is a text input field labeled "Enter name of New Campaign" and a button labeled "En". At the bottom of the form is a blue button labeled "+ Add Campaign".

Рис. 2. Створення рекламної кампанії

2. Опис вмісту веб-сторінок

Додаток містить всю інформацію на одній сторінці та має верхнє меню для швидкого доступу до них. Модуль «Dashboard» містить загальну статистику та вкладені посилання до інших сторінок форму створення нової рекламної групи (рис. 2). Модуль «Balance» містить інформаційні блоки про фінансовий статус акаунту (рис. 3).



The screenshot shows the "Balance" page. At the top, there is a breadcrumb "Home > Balance" and a status "Active Publisher (oleg+piu2@plasticjam.com)". Below this is the heading "Available balance". The main content area is divided into two columns. The left column shows the "Balance in Bitcoin" as "40.713395" and its equivalent in dollars as "\$ 319,778.61". Below this is a "Withdraw" button. The right column shows the "Top Up (฿)" section with the instruction "Send your Bitcoins to this address:" followed by a text input field containing the address "1DmUxPw9d2pHhYKqALgFSPVjP2FAFyKdiz". To the right of the input field is a QR code.

Рис. 3. Фінансовий баланс акаунту

3. Процедура авторизації користувача

Користувач може авторизуватися, ввівши логін і пароль при вході в додаток. Для незареєстрованих користувачів присутнє посилання на відповідну сторінку додатку.

Advertiser Publisher Blog B1 = \$ 7846.18 Sign up Log in

Sign In

@ Email

Password

SIGN IN

Not registered? Sign Up - It's quick and easy!

Рис. 4. Авторизація

4. Заявка на видавництво

Користувач може залишити заявку на видавництво, заповнивши відповідну форму (рис. 5).

Add New Publisher Site

Your website's information









Please note that you will be able to place advertising blocks only on sites that have been approved by the BitMedia administration. You may add as many sites as you like after your account is approved.

In case you would like your account to be activated, please specify at least one site you are planning to use for the ad publishing.

Specify Your Site's URL *

| | |
|---|----------|
|  http:// | Site URL |
|---|----------|

Category *

| | | | | | | | |
|--|--|---|--|---|--|---|--|
|  Bitcoin |  Games |  Dating |  Finance |  Shopping |  Tech |  Pharma |  Other |
| Please, select up to 5 subcategories: | | | | | | | 0/5 |
| <u>Mining</u> | | <u>Faucets and giveaways</u> | | <u>Manuals</u> | | | |
| <u>Ditcoin for beginners</u> | | <u>Wallets</u> | | <u>Blogs</u> | | | |
| <u>News</u> | | <u>Exchanges</u> | | <u>Investment</u> | | | |
| <u>Forums</u> | | | | | | | |

Site Language *

| | |
|---|---|
|  English | ▼ |
|---|---|

Description *

| |
|-------------------------------------|
| Type here short site description... |
|-------------------------------------|

☐ I agree with [Partnership terms](#) *

Add Publisher Site

Рис. 5. Заявка на издавництво